A domestic electric water heater application for Smart Grid.

by

Andrew Hector Cloete

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Research) in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr. M. J. Booysen

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2017

Abstract

The national supplier of electricity in South Africa, Eskom, is struggling to meet the country's demand and is subsequently operating at a historically low reserve margin. Domestic electric water heaters (EWHs) are some of the largest consumers of electricity in the residential sector. EWHs are also responsible for high demand peaks, which add additional strain to the electrical grid. Currently, redistributors manage EWHs countrywide by using ripple control to remotely switch EWHs. However, ripple control is a unidirectional system that switches EWHs in bulk and subsequently cannot take individual customers into consideration. Peak demand management of electricity supply often disregards the comfort level of EWH users.

This study presents advancements on two separate, but complementary, components that are necessary in a system in order to address both aspects, i.e. demand-side management as well as comfort levels. The components include a bidirectional communication system and computationally efficient EWH model. The former has the ability to connect a large number of EWH for monitoring and control. The latter is an accurate prediction tool to assess the EWH state in order to assist in its switch control.

For the purpose of this research, two communication systems were developed for existing EWH monitoring that are equipped with cellular modems. The first system complies with the recent specifications released for machine-to-machine (M2M) communication known as *SmartM2M*. The second system uses popular web technologies and employs the lightweight MQTT application layer protocol. Both systems functioned as designed. It was found that the *SmartM2M* standards, although effective, lacks adoption at this point in time which made application development cumbersome. MQTT was employed by using community developed software for both client and server side, and proved highly effective. The cellular modem interface (known as the extended AT command set) proved challenging, as it is an obscure, vendor dependent interface with little community support. A communication software stack was developed that systematically interfaces with the modem. The communication software stack proved highly effective, enabling the MQTT protocol, the dialling of network service codes and remote firmware updates.

A rigorous laboratory experiment was developed to validate the accuracy of computationally efficient EWH models proposed in a previous study. The experiment entailed performing automatic, consistent and precise water draw-off and heating, while periodically measuring several metrics. Eight datasets of 9- day experimental data was generated and used to compare with corresponding simulated results using the EWH nodal models. Overall, the models showed good accuracy in predicting input energy, energy efficiency and water draw-off temperature. Under scheduled heating control (as opposed to "always on"), increased model accuracy was observed in all three aforementioned categories. The experimental data was also used to show that the energy savings possible by implementing a heating schedule is approximately 6%, given the same effective output energy as when an "always on" heating schedule is applied.

Uittreksel

Suid Afrika se nasionale elektrisiteitsverskaffer, Eskom, swig toenemend onder die land se elektrisiteitaanvraag. Gevolglik word die kragnetwerk teen ‘n historiese lae reserwemarge bedryf. Huishoudelike elektriese warmwater verhitters (EWVs) is van die grootse elektrisiteitsverbruikers in die residensiële sektor. EWVs is verantwoordelik vir hoë aanvraag pieke, wat dus bykomende druk op die kragnetwerk plaas. Op die oomblik gebruik herverspreiders rimpeleffekbeheer om EWVs oor ‘n afstand te aan of af te skakel. Die geriefsvlakke van EWVs-gebruikers word dikwels laag geprioritiseer in ‘n poging om piekaanvraag van kragvoorsiening te bestuur.

Hierdie studie bied verbeteringe op twee afsonderlike, maar komplementêre, komponente wat ‘n stelsel benodig om beide aspekte aan te spreek, m.a.w. aanvraagbeheer en geriefsvlakke. Die komponente sluit ‘n tweerigting kommunikasiestelsel en ‘n berekeningseffektiewe EWH model in. Eersgenoemde het die vermoë om ‘n groot aantal EWVs te verbind ten einde hulle te monitor en beheer. Laasgenoemde is ‘n akkurate voorspellingsinstrument om die EWV-toestand te assesseer ten einde skakelbeheer te help fassiliteer.

Vir die doel van hierdie navorsing, is twee kommunikasiesstelsels vir bestaande EWV-moniterings hardeware (toegerus met sellulêre modems) ontwikkel. Die eerste stelsel voldoen aan die hedendaagse spesifikasies vir masjien-tot-masjien (M2M) kommunikasie wat bekend staan as "*SmartM2M*". Die tweede stelsel gebruik populêre webtegnologieë en benut die liggewig MQTT-toepassingsvlakprotokol. Beide stelsels funksioneer volgens die beplande ontwerpspesifikasies. Daar is gevind dat die *SmartM2M* standaarde effektief is, maar ‘n gebrek aan grootskaalse aanvaarding, het implimentering en toepassing daarvan vermoeilik. The MQTT-protokol is geïmplimenteer deur gemeenskapsgedrewe sagteware vir beide the kliënt- en bedienertoepassing. Implementering van die MQTT-protokol was hoogs doeltreffend. Die sellulêre modem koppelvlak (bekend as die "*AT*" bevelstel) het vele ontwikkelingsuitdagings gebied aangesien dit ‘n onbekende, verkoper-afhanklike koppelvlak is, waavoor daar minimale gemeenskapsondersteuning beskikbaar is. ‘n Kommunikasie sagtewarestapel, wat stelselmatig met die modem koppel, is ontwikkel. Hierdie sagtwarestapel was hoogs doeltreffend om die MQTT-protokol te aktiveer, skakeling van netwerkdienskodes te hanteer en die afstands-hegware op te dateer.

‘n Nougesette laboratoriumeksperiment is ontwikkel om die akkuraatheid van die berekenings-effektiewe EWV modelle (uit ‘n vorige studie) te bevestig. Die eksperiment het outomatiese uitvoering van bestendige, noukeurige wateraftapping en verhitting, met periodiese metings van verskeie parameters, behels. Deur die EWV nodale modelle te gebruik, is agt datastelle (bestaande uit 9-dae experimentele data) gegenereer. Dit was gebruik om eksperimentele data met ooreenstemmende gesimuleerde data te vergelyk. In die geheel, het die modelle goeie akkuraatheid getoon om intree-energie, energie-doeltreffendheid en gemiddelde wateraftrek-temperatuur te voorspel. Onder geskeduleerde verwarmingsbeheer (in teenstelling met "altyd aan" beheer) is verbeterde mode-

lakkuraatheid in al drie die bogenoemde kategoriee waargeneem. Die eksperimentele data was ook gebruik om aan te toon dat sowat 6% energiebesparing vir dieselfde uitsetenergie moontlik is (teenoor ‘n "altyd aan" skedule) deur ‘n verhittingskedule te implimenteer.

Publications

The work in this manuscript has been accepted for publication as follows:

- A.H. Cloete and J.W.K. Brown and M.J. Booysen and R. Steinke and T. Magedanz, "Smart Grid Application using ETSI M2M: Domestic Electric Water Heaters," *SAUPEC 2016 Proceedings*, January 2016.
- M.J. Booysen and A.H. Cloete, "Sustainability through Intelligent Scheduling of Electric Water Heaters in a Smart Grid," in *Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/- CyberSciTech), 2016 IEEE 14th Intl C.* IEEE, 2016, pp. 848-855.
- M.J. Booysen and A.H. Cloete and R.C. Sandell, "Smart electric water heaters: A system architecture proposal for scalable IoT," in *Internet of Things, Data and Cloud Computing, 2nd Intl Conf (ICC 2017)*, 2017.

Acknowledgements

I would like to thank the following people and organisations for their invaluable support over the past two years:

- My supervisor, Dr Thinus Booysen for his leadership and also with presenting me with many opportunities for personal and professional growth.
- The Water Research Commission for their research grant, facilitating the expansion of the scale at which this study could be carried out.
- MTN for their continued support and funding through the MTN Mobile Intelligence Lab.
- My colleagues and fellow students involved in contributing to the development of the second communication system, which provided the necessary dependencies to carry out this study. Special thanks to Nico Naude, Robert Sandell, Lourens Visagie, Marcel Roux, Jonathan Brown and Philip Nel.
- The Department of Electrical and Electronic Engineering for their technical support especially in providing the necessary trade skills and labour for installing the EWH laboratory setup.
- Elsabé Cloete for reviewing this manuscript and patiently pointing out grammatical errors.
- My parents for their continued love and support.
- Family and friends for their emotional support and encouragement.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms and abbreviations

AMI	Advanced Metering Infrastructure
API	Application Programming Interface
CoAP	Constrained Application Protocol
CRUD	Create, Retrieve, Update, Delete
DR	Demand Response
DSM	Demand-Side Management
ETSI	European Telecommunications Standards Institute
EWH	Electric Water Heater
GET	Refers to the *get* HTTP method
GPIO	General Purpose Input/Output
GSM	Global System for Mobile Communications
H2H	Human-to-Human
IMEI	International Mobile Equipment Identity
IoT	Internet-of-Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
JSON	Java-Script Object Notation
LTE	Long-term Evolution
M2M	Machine-to-Machine
MQTT	Message Queuing Telemetry Transport
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
OM2M	Open M2M
OTA	Over the Air
PDP	Packet Data Protocol
POST	Refers to the *post* HTTP method
PPP	Point-to-Point Protocol
REST	REpresentational State Transfer
RPi	Raspberry Pi
SABS	South African Bureau of Standards
SC	Schedule Control
SCADA	Supervisory Control and Data Acquisition

SDO	Standards Developing Organization
SG	Smart Grid
SRC	Solicited Response Code
SRM	Solicited Response Message
TC	Thermostat Control
TCP	Transmission Control Protocol
UART	Universal
UDP	User Datagram Protocol
UR	Unsolicited Response
URL	Uniform Resource Locater
USSD	Unstructured Supplementary Service Data
USSD	Unstructured Supplementary Service Data
WSN	Wireless Sensor Networks
XML	eXtensible Markup Language
xSCL	(any) Service Capability Layer

Constants

c	Specific heat capacity of water ..... 4184 Joule/kg.°C
ρ	Density of water ..... 1000 kg/m^3

List of Symbols

R	Total thermal resistance of and EWH cylinder ..... [°C/Watt]
k	Exponential cooling constant ..... [1/sec]
T_{out}	Temperature measured at the water outlet of the EWH, on the pipe exterior surface ..... [°C]
T_{in}	Temperature measured at the water inlet of the EWH, on the pipe exterior surface ..... [°C]
T_{amb}	Temperature measured in the air at roughly 300mm away from the EWH surface [°C]
V_{cyl}	Fixed internal volume of water in an EWH cylinder ..... [m^3]
T_{cyl}	Average temperature of the fixed internal volume of water in an EWH cylinder [°C]
U_{cyl}	Enthalpy of the fixed internal volume of water in an EWH cylinder .... [Joule]
V_{draw}	Volume of water drawn-off from an EWH cylinder ..... [m^3]
T_{draw}	Average temperature of the volume of water drawn-off from an EWH cylinder [°C]
Q_{draw}	Enthalpy of a volume of water drawn-off from an EWH cylinder ..... [Joule]
Q_{loss}	Energy loss of an EWH cylinder due to thermal radiation ..... [Joule]
Q_{elec}	Electrical energy supplied to the EWH via electric element ..... [Joule]
P_{elec}	Instantaneous power supplied to the EWH via electric element .... [Joule/sec]
η	EWH energy efficiency ..... [N/A]

Chapter 1

Introduction

According to the Global Energy Statistical yearbook, the total world wide consumption of domestic energy in 2015 was just over 20 000 Terra Watt hours (TWh) [1]. This is a 54% increase in consumption from the year 2000. Specifically in Africa, the consumption of domestic energy in 2015 was 621 TWh, a 64% increase from the year 2000. Figure 1.1 shows the yearly electrical energy generated in six continents from 1985 to 2015 (according to BP electricity review 2016 [2]). According to BP, the average yearly increase in electricity supply over the past ten year has been 2.8%.

In South Africa robust economic growth (especially in the residential sector) between 1994 and 2006 has placed demanding pressures on the national electricity provider (Eskom) [3]. According to the authors of [3], the electricity reserve margin decreased from 25% to 20% between the years 2002 to 2004, and again to 16% in 2006. The minimum reserve margin is normally set at 15%. In 2008, Eskom started implementation of rolling blackouts (load shedding) in order to stabilise the grid and mitigate the risk of a complete blackout [4]. Along with implementing load shedding, Eskom responded with several impromptu demand-side management (DSM) projects such as distributing millions of free compact fluorescent lights and advertising energy tips as financial incentives to lower overall consumption [5]. In their 2015/16 annual report, the South African Department of Energy (DoE) stated that the country is still suffering from severe shortages of electricity supply [6]. Currently the DoE is executing a "Five Point Plan" to mitigate the challenges that includes increasing capacity through Independent Power Producers (IPPs), which aims to add an additional 17 000 MW capacity to the grid by the end of 2022. It also includes further DSM options for South Africans [6].

1.1 Smart Grid

Smart Grids (SGs) aim to address concerns about energy by improving the overall efficiency of power grids through means of a pervasive intelligent communication network. Traditionally, most national electricity is generated and distributed using a centralised, hierarchical power grid. This power grid can generally be divided into four distinct parts, *generation*, *transmission*, *distribution* and *customer*, and is arranged as shown in Figure 1.2 where power flows *unidirectionally* from generation to customer [7]. This architecture usually lacks an appropriate communication infrastructure to provide the system with relevant real-time feedback. This implies that the system essentially operates in "open-loop" mode [8]. Utility providers base load demand on estimates by using historical data and they always need to be prepared to meet *peak* demand within a safe margin of op-

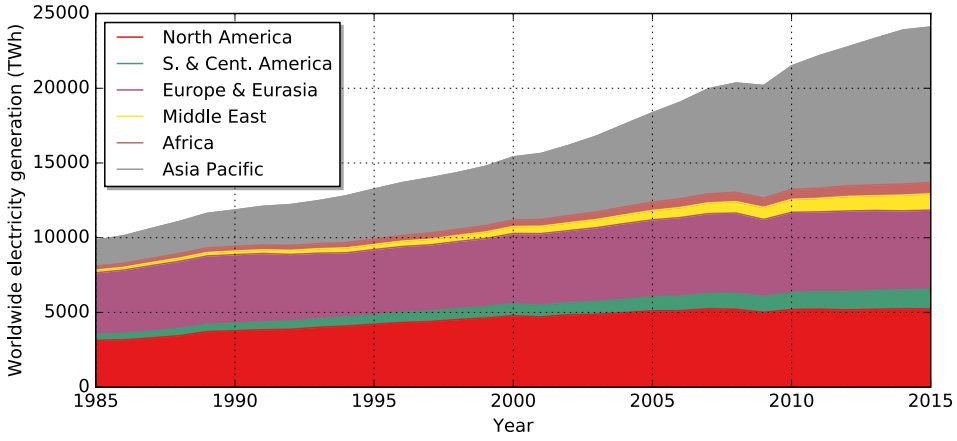


Figure 1.1: Global energy supply per year with an average of 2.8% increase per year over the past 10 years. Data available, see [2].

eration. Generation supply capacity is typically 20% above peak demand, which is only used 5% of the time [9]. The main objective of Smart Grid is therefore to increase the efficiency and reliability of a power grid by adding the appropriate infrastructure that "closes the loop". As will be shown, equipping a power grid with the necessary sensory and communication infrastructure has far-reaching benefits for both provider and customer.

The notion of Smart Grid (SG) is vast and has yet to see a single, all-encompassing definition. One of the leading authorities on SG is the *National Institute of Standards and Technology (NIST)*. NIST currently appears to hold the most comprehensive framework and road map specifications for Smart Grid interoperability standards [10]. NIST attempts to define the concept of SG by using a list of distinguishing characteristics. Some key points quoted from [10] are as follows:

- *"Increased use of digital information and controls technology to improve reliability, security, and efficiency of the electric grid"*;
- *"Deployment and integration of distributed resources and generation, including renewable resources"*;
- *"Development and incorporation of demand response, demand-side resources, and energy- efficiency resources"*;
- *"Integration of "smart" appliances and customer devices"*.

According to the authors of [11] the European Union has defined SG as:

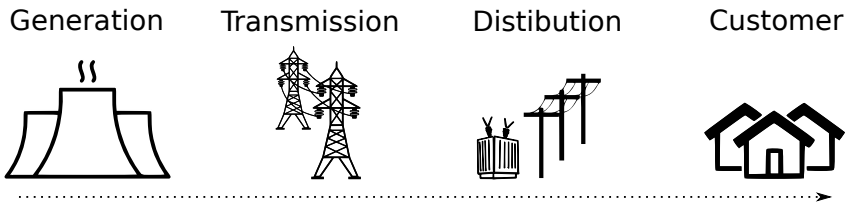


Figure 1.2: Concept of the traditional grid showing unidirectional flow of electricity [7]

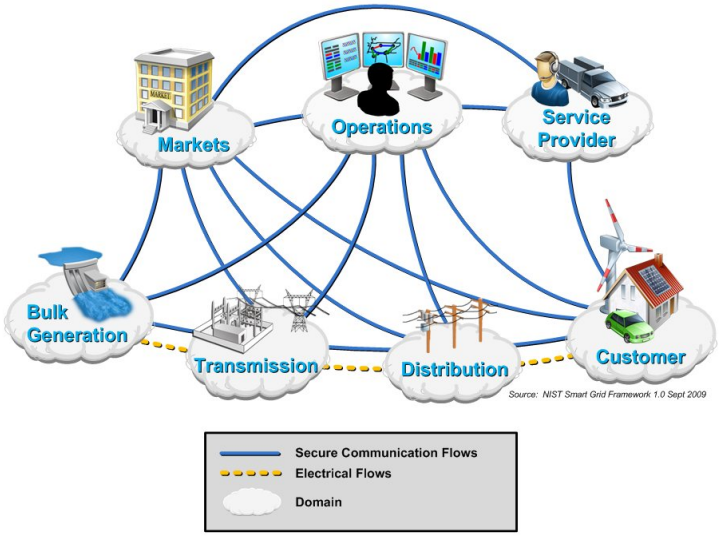


Figure 1.3: Conceptual model of the Smart Grid. Extracted from NIST SG framework [10].

"an electricity network that can cost-effectively integrate behaviour and actions of all users connected to it – generators, consumers and those that do both – in order to ensure economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety".

Figure 1.3 presents the envisioned SG framework by NIST, showing the interaction of actors in different domains. Comparing Figure 1.3 to the traditional grid in Figure 1.2, it can be seen that the same four energy actors are shown in both. However, in contrast with the traditional grid, where the boundaries between the actors are well defined, the architecture of SG is decentralised and allows actors to take on multiple roles. Apart from the traditional role players, NIST defines three other role players that participates explicitly in the SG architecture. These are (1) markets, (2) operations, and (3) service providers. Market actors maintain the balance between supply and demand of electricity. This consists of retailers, bulk suppliers, traders, *et cetera*. The operations domain ensures optimal operation of the transmission and distribution domains though careful monitoring and system analysis. Service providers manage the delivery of electricity to customers and perform tasks such as billing and customer account management. In order for any of these actors to operate efficiently, a real-time communication system is required [12].

Traditionally, the communication systems found in the electric grid are grouped under the umbrella term of *Supervisory Control and Data Acquisition (SCADA)*. SCADA systems are generally proprietary and built to suit only a narrow range of industrial type applications. Current SCADA systems do not meet the requirements for a fully decentralised and autonomous control communication network [13]. An SG communication system may be broken into two distinct parts. Firstly, a physical network that interconnects all smart meters, sensors, control centres, *et cetera*. This is typically characterised

by requirements such as network latency, data delivery critically, reliability, security and time synchronisation. Secondly, a middleware software system is required that will allow the heterogeneous networks and devices to be interfaced in a uniform and consistent manner. The typical characteristics of middleware is interoperability, scalability and autonomy.

1.2 IoT and M2M as enablers for Smart Grid

The efforts towards a standardised communication infrastructure for power grids is only one example of a greater world-wide trend to establish a pervasive communication system in all domains. This paradigm, which envisions connecting a large variety of *objects* or *things* to the Internet by means of sensors and actuators for large scale interoperability, information exchange and intelligence, is generally referred to as *The Internet of Things* or *IoT* [14].

According to Cisco [15], the number of devices connected to the Internet exceeded the human population somewhere between 2008 and 2009, after an explosive growth spurt from only 0.08 devices per person in 2003. Cisco estimates that by 2020 around 50 billion devices will be connected to the Internet exceeding the human population about 6.5 to 1. Traditionally, the Internet generally consists of, and is optimised for voice calls, messaging and web browsing. This is sometimes referred to as *human-to-human (H2H)* communication. However, with the rise of IoT, Internet traffic is increasingly being characterised by sensors that autonomously generate data not intended for human consumption. This traffic is subsequently known as *machine-to-machine (M2M)* communication. The architecture of M2M is significantly more intricate than that of H2H, since it requires an increase in the level of automation so that systems and devices can share and exchange information without human intervention [16]. Developments in communication technologies for IoT are inevitably intertwined with the developments for Smart Grid. In fact, M2M standards developing organisations (SDOs) recognise that Smart Grid is a key subsystem of IoT [17]. This means that in order to develop Smart Grid applications, a thorough understanding of the IoT and M2M landscape is needed. The requirements, architecture and, more recently, the specifications for M2M, are addressed in detail in this work.

1.3 Background: Domestic water heating control in South Africa

According to Eskom, the residential sector uses about 17% of the total energy generated in South Africa. Furthermore, during peak periods (7 AM to 10 AM in the morning, and 5 PM and 9 PM in the evening), the residential sector amounts for up to 35% of the total energy demand. This sharp peak is attributed to household electric water heaters (EWHs), of which there are estimated to be over 5.4 million in South Africa [18].

The first control scheme of EWHs in South Africa started in Benoni in 1956 and in Sasolburg in 1959 [19]. The main reason for the scheme was to reduce the cost of electricity to the municipalities, which redistribute from the electricity supply industry. Although

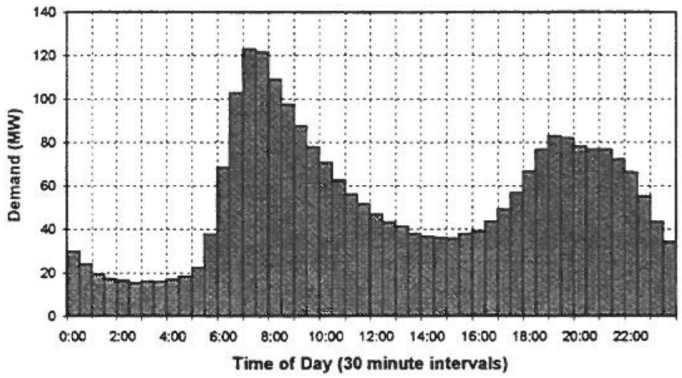


Figure 1.4: Eskom notch test results from May 1997 showing average demand of roughly 120000 EWHs over 24 hours. Excerpt from [24].

households in South Africa pay a time-independent rate for electricity, municipalities are charged by a time-of-use (TOU) tariff scheme called *MegaFlex*, which is significantly more expensive during peak demand periods [20] [21, p. 25]. The remote switching of EWHs is achieved by installing a *ripple control* unit at each household. A ripple control unit is a relay device that can be activated by transmitting a digital signal modulated onto the power lines (radio is also used in some cases). The concept of ripple control dates as far back as 1928 and is used extensively throughout the world [19, 22, 23].

In preparation for improving Demand Side Management (DSM) programs in South Africa, Eskom carried out a milestone experiment in 1997 called the *"notch test"* program that collected baseline data on domestic water heating [19, 24]. The principle of the notch test program was to bulk switch the installed ripple control units throughout South Africa for brief periods of time during the day and then measure the effect the switching has on the grid. By measuring the "notch" in the grid load that was created by the switching, Eskom could calculate a 24-hour load profile of domestic water heating. The program was carried out over three seasons in twelve municipalities and included an estimate of 120 000 EWHs [24]. Figure 1.4 shows the 24-hour load profile of the average weekday in May 1997. The figure depicts two load peaks during the day, namely a large peak in the morning around 7 AM, followed by a smaller peak at around 7 PM.

Eskom's motivation for installing ripple control units onto EWHs makes sense when considering Figure 1.4: An EWH is a high-load household appliance that is prone to periodic load peaks due to the synchronism of most domestic routines. However, since an EWH can effectively be used to store energy for later use, it is the ideal candidate to shift grid load to achieve what is known as *peak-shaving and valley-filling*. Since the ripple control scheme is based on *average* load, it has a major flaw since it may cause *individual* households to lack hot water when it is required. This is known as the *customer comfort* constraint and is neglected by ripple control since it is a *unidirectional* communication system, unable to account for individual EWH temperature.

1.4 Thesis problem statement and objectives

This section presents the main problem statement that will be examined in this work, as well as the objectives identified to systematically address the problem.

1.4.1 Problem statement

An EWH is (in most cases) considered to be the single largest consumer of electricity in the residential sector. Their aggregate load on the electric grid is intensified by being prone to periodic behaviour of load peaks and valleys. However, since EWHs can *store* energy for delayed use, the aforementioned characteristics render them ideal candidates for grid load shifting schemes. In fact, ripple control is used throughout the world to perform this task. But since ripple control is a unidirectional communication system that can only switch EWHs in bulk, it cannot guarantee that any individual household will have hot water when it is needed. To ensure that the customer satisfaction is always met, a *bidirectional* communication system is required that can measure the state of each individual EWH. Moreover, not only is a measurement of the current state of the EWH needed for customer satisfaction, a prediction of the near-future state is also required for informed decisions on whether or not heating is required. In other words, some basic modelling of each EWH is also required to estimate its effective energy available for use.

1.4.2 Proposed solution and objectives

The above problem is addressed in this study through two separate, but complementary parts. The first, is to develop an appropriate bidirectional communication system for the monitoring and controlling of EWHs. Such a system must conform to the *status quo* of current Smart Grid and M2M/IoT technological developments. The second, is to perform laboratory experiments to collect precise and consistent EWH measurement, which may be used to validate previously proposed EWH models. Accurate models are required for EWH state prediction. Additionally, EWH energy efficiency is also investigated to provide insight into the most effective control strategies in terms of overall energy reduction and peak demand management. Three main objectives have been identified for this work, and are listed below, along with their appropriate sub-objectives.

Objective 1: Develop a bidirectional communication system

Develop an M2M system that enables bidirectional communication of real-time sensing information and access control for EWHs, adhering to the requirements listed below.

1(a): The system must conform to the Smart Grid and M2M/IoT vision of a distributed, decentralised architecture with horizontal interoperability.

1(b): Cellular access networks must be used to establish communications to multiple existing EWHs. New EWH hardware controllers will not be developed in this work. The system must use the existing hardware controllers developed by Brown [25] (discussed in the next section). These hardware controllers are only equipped with cellular modems for communication.

1(c): The EWH hardware controllers must both report periodic measurements, and be able to receive commands and respond with an action such as switching the EWH element.

Objective 2: Validate the Nodal models presented by Nel [26]

An energy model that is computationally inexpensive and applicable for large-scale SG deployment has recently been proposed by Nel [26]. The proposed model also takes into account whether the EWH is mounted vertically or horizontally - a relevant factor in SA. However, this model has not been validated rigorously to determine if it is sufficiently accurate, given a known water draw-off routine and heating schedule. The second aim of this work is therefore to validate the model's proposed by Nel [26]. This will be achieved through the following sub-objectives.

2(a): Design an experiment to validate the EWH models. This entails designing a realistic water draw-off routine and heating schedule.

2(b): Implement a laboratory setup to execute the above experiment repeatedly and consistently.

2(c): Determine the physical model parameters of the experimental EWH. This includes the thermal radiation constant and thermostat set-point.

2(d): Implement the EWH models in software and build a platform to simulate EWH output using the data from the experiments in *2.b*

2(e): Test the models for both *thermal* accuracy and *operational* accuracy. The former considers the accuracy given the *exact* input energy and water draw-off. The latter considers the accuracy given only the water draw-off and uses a *simulated* thermostat for input energy. This is an important distinction, since only the latter can be used for *prediction* purposes.

2(f): Determine if the model displays an inclination to be more accurate for one particular type of heating control i.e. "always-on" heating versus scheduled heating.

2(g): Determine if the model displays an inclination to be more accurate for one particular EWH orientation i.e. horizontal versus vertical orientation.

Objective 3: Determine the energy savings possible using basic scheduling

A bidirectional communication system that is coupled with an accurate EWH model may be used for peak-shaving and valley-filling while maintaining customer comfort. A follow-up research question is to determine the *overall* energy savings possible by implementing a heating schedule that *matches* a household warm water usage profile (as opposed to always supplying the EWH thermostat with electricity). Some literature states that up to 16% energy can be saved by implementing a heating schedule [27]. However, as with the model validation, this lacks rigorous experimentation.

3(a): Design an experiment that can be used to measure EWH energy efficiency. This entails designing an appropriate water draw-off routine and heating schedule.

3(b): The heating schedule may not violate the customer comfort constraint. For this study, the minimum water draw-off temperature is set at 49°C [28].

3(c): Determine the conditions required to make a valid comparison between thermostat control (TC) and schedule control (SC).

3(d): Determine the difference in efficiency between a *vertically* and a *horizontally* mounted EWH.

1.5 Foundational work

This section presents a short overview of the project as a whole and the context in which it was carried out. The goal is to provide the reader with the necessary backdrop to appreciate the design decisions.

The project forms part of a greater project group that aims to deploy *Smart* EWH controllers on a large scale in South Africa. The project fundamentally aims to coalesce the research presented in two previous studies (that is [25] and [26]) in order to provide further progress towards a fully operational end-to-end system suitable for large scale deployment. Specifically, Brown [25] presents sensory and controller hardware that can be fitted to an EWH to measure and control temperatures, power consumption and water consumption. This work is advanced by equipping the hardware device with M2M functionality along with the appropriate back-end system as envisioned by SG and IoT. Nel [26] derives an energy model that may be used for EWH state prediction on a large scale. His work is advanced by validating the results proposed through performing rigorous laboratory experiments. This is a necessary task since an accurate model of the EWH is required to perform large scale EWH state prediction and control while at the same time satisfying the human comfort constraint.

Another important aspect of the project context is that the design requirements changed from *research prototype* to *pilot product* one year of after project initiation. The initial goal was to install at least 10 EWH prototype units to volunteer households in the Cape Winelands district. Due to additional funding (from the *Water Research Commission (WRC)* [29]) the goal of the project changed to install 450 units in the local municipality of Mkhondo [30] in Mpumalanga, South Africa. The requirements were revised to account for the additional challenges and risks involved at an increased scale. In this work, the former system will be referred to as **System A** and the latter as **System B**. Both systems are presented in contrast with each other as each highlights a different aspect of M2M. The focus of System A is *standard compliance* and *interoperability*. For System B, the focus is efficient *protocols* and *scalability*. The increase in project size also meant that additional developers were brought on-board to manage the work load, which is inevitably included in this work. Where the work is not that of the author, it is clearly denoted.

1.6 Contributions

This paper makes the following contributions:

- Presents a practical implementation of the SmartM2M standards, which may serve as an example for a newcomer to interpret the formal standards specification documentation (System A).

- Presents a practical implementation of using the AT command interface of cellular modems for which there is little community support (System B).
- Delivers 8 datasets of 12 days periodic experiments on EWHs sampled at one second intervals and stimulated by a precise draw-off routine and heating schedule.
- Delivers 4 datasets of EWH cooldown data of a period of 5 days sampled at one second intervals.
- Delivers 2 datasets of EWH static heating over a period of 5 days sampled at one second intervals.
- Uses the above experimental data to accurately estimate EWH model parameters and then validate the models to show acceptable accuracy.
- Uses the above experimental data to quantify the benefit of applying a heating schedule to EWHs and illustrates that other works tend to be optimistic about the possible energy savings.

1.7 Scope of Work

The scope of this study is restricted by the following barriers:

- New software implementations of the SmartM2M standards are not developed. Only existing implementations of the standards are used (specifically the OM2M platform). Furthermore, only open-source and other freely available software are used to develop the M2M systems.
- Since the M2M systems are regarded as research prototypes, major steps to ensure cyber security are not taken. However, this does not imply that this study does not regard cyber security essential for commercial applications.
- The M2M systems are presented as-is. Quantitative comparisons with other M2M systems are not performed.
- Existing EWH hardware controllers are used. New hardware controllers are not developed in this work.
- All EWH experiments are conducted on EWHs that are installed according to SABS specifications. Experiments which compromise EWH integrity are not considered.
- The existing EWH thermal models that are evaluated in this study are used as-is and not modified based on simulation results. That is, no model development is performed.

1.8 Thesis structure

Chapter 2 presents a literature review on three domains applicable to the thesis objectives. In the first part, the concept of Smart grid is described. This forms the basis of the communication architecture required for decentralised generation, transmission, distribution and consumption. In the second part, the M2M communication domain is

described by reviewing the recent *SmartM2M* standards and lightweight application layer protocols. In the third part, thermal modelling of EWHs is reviewed. The EWH nodal models presented by Nel [26] are discussed in depth. This chapter concludes with an overview of the EWH hardware controllers by Brown [25], with particular focus on the AT command interface for cellular modems.

Chapter 3 describes the design and implementation of two M2M communication systems, namely System A and System B. System A focuses on the middleware layer of M2M and conforms to the SmartM2M standards. The REST architecture of the system is discussed in depth. System B focuses on lightweight protocols and scalability by implementing the MQTT protocol. The communication software stack of the EWH hardware controller is presented in detail.

Chapter 4 presents the results of the two M2M systems. The efficacy and anticipated functionality of the two systems are presented. Evaluation of four key points, namely: the SmartM2M standard, the OM2M platform, the MQTT protocol and the AT command interface, is described.

Chapter 5 describes the design and implementation of the EWH lab experiment for EWH model validation and efficiency evaluation. Firstly the experiment is defined and includes a water draw-off routine and heating schedule. Secondly the physical setup is presented for performing autonomous, precise and consistent EWH stimulation while recording several metrics on a one second interval.

Chapter 6 presents the results of the EWH experiments, as well as the simulated results using the experiment's data. Firstly the EWH model parameters (such as thermal resistance) are determined using a some additional experiments. The results focus on three main metrics namely, input energy, energy efficiency and mean water draw-off temperature.

Chapter 7 concludes the study by evaluating the three objectives of Section 1.4.2. Finally, recommendations for future work regarding both the M2M systems and the EWH laboratory experiments are presented.

Chapter 2

Literature Survey

A large-scale monitoring and control system for EWHs lies at the intersection of three domains, namely, (1) Smart Grid, (2) IoT and (3) EWH modelling, as shown in Figure 2.1. This chapter provides a survey of the relevant literature for the three domains. To the best of the authors' knowledge little literature exists on a full end-to-end EWH monitoring and control system design for Smart Grid application complying to M2M standards.

This chapter presents an overview of SG with its strategies and architecture. This is followed by a discussion on the recent efforts towards M2M standardisation. Focus is placed on the current state of available technologies so as to motivate the design decisions in Chapter 3. Next, an overview of EWH modelling is presented with a detailed summary of the model proposed by Nel [26]. Finally an overview of the EWH hardware by Brown [25] is given, specifically focussing on the communications capabilities.

2.1 Smart Grid overview

As introduced in Chapter 1, the concept of Smart Grid can be broken into two inquiries. First the *"what?"* looks at the specific strategies needed to optimise a power grid. Examples of such strategies include Demand Response (DR) programs, Energy Management Systems (EMS) and Virtual Power Plants (VPP) [13]. Secondly the *"how?"* looks at the communication infrastructure required to materialise the strategies above. This is the

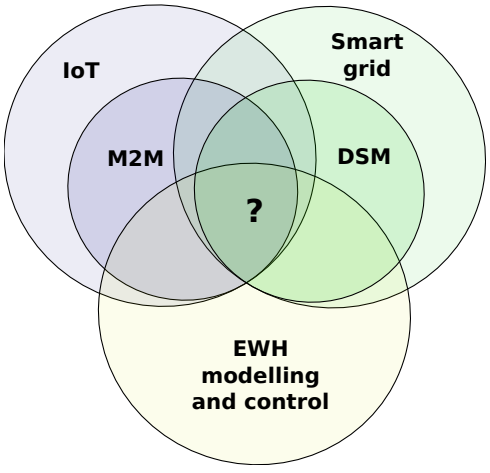


Figure 2.1: Intersection of domains required to address objectives.

system of networks and protocols that enables connectivity and information distribution amongst grid sub-systems and end-point devices. Such a system needs to be reliable, timely, secure and scalable [13].

2.1.1 Categorising Smart Grid strategies

Demand-side management (DSM) refers to the administration of the consumers of electricity in an electrical network to best suit the generation capabilities of that network. This may range from replacing incandescent light bulbs with LED lighting, to installing advanced grid load management systems [31].

In [32] DSM applications are categorised as (i) *centralised* and (ii) *decentralised*, depending on the location of the control decision. Centralised control is applied in normal conditions to meet demands and ensure efficiency. Decentralised is usually reserved for abnormal or emergency conditions, such as generator failure. In the latter case, a control decision is performed by each device, based on its immediate environment. For example, a high load device (such as an EWH) can switch its element off if it detects a voltage dip. In [33], heating, ventilation and air conditioning (HVAC) systems in a home environment are simulated and it is demonstrated that a healthy Smart Grid would consist of both centralised and decentralised control methodologies.

Whereas in the past DSM is usually driven by the utility provider, more recent strategies aims to involve the customer directly. This is known as *Demand Response (DR)* and refers to providing customers with incentives to alter their behaviour as needed. These incentives are (in most cases) financially based [34]. In [9] a review of the current state of residential DR strategies is presented. The paper further categorised DR schemes into (1) *direct* and (2) *indirect* schemes. Direct schemes give the utility provider some degree of authority to schedule, reduce or switch appliances. Indirect control involves handing control over to the customer, but providing incentives to comply to the needs of the provider. In this way users can balance *minimum payment* versus *maximum comfort*.

2.1.2 Smart grid communication

The scope of an SG communication system is vast and cannot be covered in detail in this work. Only a brief overview is presented here. As stated in the Introduction, a communication system for SG may be broken into two parts, (1) a communication infrastructure, and (2) a middleware platform [13]. The communication infrastructure consists of the networks, sensors, protocols, computers and so on, which present the physical interconnected system. The middleware platform is the software layer or virtual architecture of the network and, as the name suggests, lies between the business applications and the communication infrastructure of the SG. The middleware provides functions such as data storage, device addressability, security, standard application interfaces and management capabilities.

2.1.2.1 SG communication architecture

The SG communication infrastructure is expected to connect a large number of devices, by using a wide variety of communication technologies. Surveys on SG communication architecture are presented in [7, 12]. The IEEE 2030-2011 [35] standard is currently a widely accepted guideline for SG communication. It proposes a system of a hierarchical

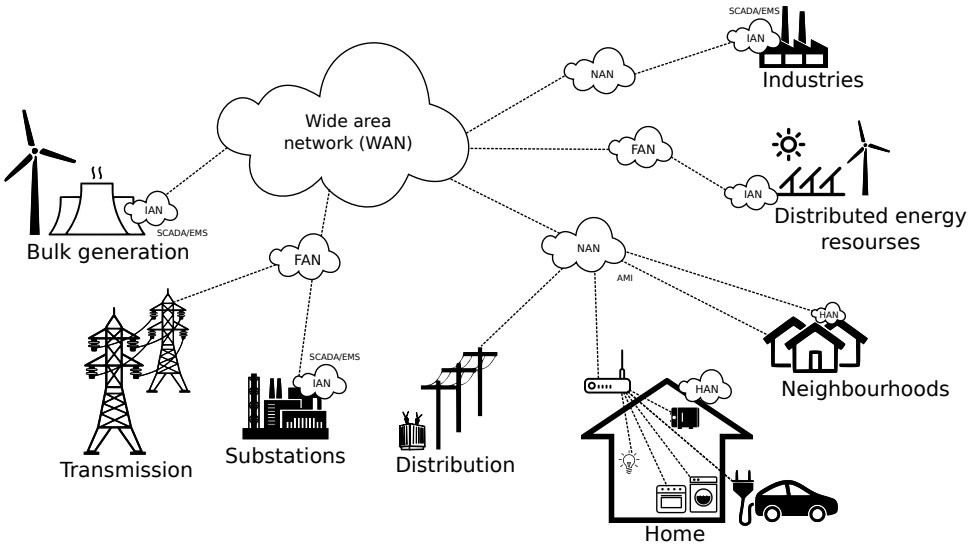


Figure 2.2: Smart Grid communication network architecture showing the concepts of private area-, wide area- and core network. Adapted from [35, 10, 7].

or multi-tier architecture consisting of increasingly narrower sub-networks. This concept is depicted in Figure 2.2. As shown, the architecture consists of *private networks*, *neighbourhood/field area networks* and *wide area networks* [8, 35].

Private networks: Private area networks are the so-called "last mile" networks that connect end-point customer devices. According to the authors of [8], private networks can be classified into three categories: (1) home area networks (HANs), (2) building/business area networks (BANs), and (3) industrial area networks (IANs), as shown in Figure 2.2. HANs involve the customer on a individual basis and is used to connect household appliances to the SG network. At the heart of the HAN is the concept of the *smart meter*, which is used to record consumption and receive commands from the utility such as pricing information or control switching. BANs are similar to HANs in that they cover roughly the same physical area. However, they are envisioned to constitute more connections and deliver SG services different to those applicable in domestic areas. The same underlying communication technology of HANs is therefore applicable to BANs.

IANs refer to the local networks used for industrial monitoring and control. This is not only applicable to energy intensive customers, but also to the the grid itself, such as substation and generation plants. Traditionally, this type of network is referred to a *Supervisory Control and Data Acquisition (SCADA)* system and constitutes a centralised computer-based network that consists of Remote Telemetry Units (RTUs), Programmable Logic Controllers (PLCs) and Human-Machine Interfaces (HMIs). However, according to the authors of [13], SCADA systems are not suitable for fully decentralised and autonomous nature envisioned for SG. Instead, SCADA systems may be considered to fall under the SG umbrella term *Energy Management System (EMS)* [7].

The typical bandwidth requirements for SG applications in this sub-network is around 1-100 kbps and is expected to cover ranges between 1-100 meters [36, 37]. According to [7], wireless communication is preferred over wired communication for private networks since it allows flexibly rearranging devices. Popular wireless technologies for private networks are Zigbee (IEEE 802.15.4), WLAN (IEEE 802.11), Bluetooth (IEEE 802.15.1) and unlicensed 900 MHz Industrial, Scientific and Medical (ISM) band [38].

Neighbourhood/Field area networks: This network tier clusters several private networks and aggregates the communication, providing a wider scope of situational awareness. Conceptually, NANs aggregate the residential type networks such as HANs and BANs, whereas FANs aggregate IANs. The SG term *advanced metering infrastructure (AMI)* is often found in literature to characterise these sub-networks [8, 13, 39, 40]. The purpose of AMI is to (1) monitor power quality, (2) connect and monitor individual household smart meters, (3) administrate dynamic pricing or scheduling of residential loads, and (4) estimate energy produced/stored in distributed energy resources. A comprehensive review of AMI can be found in [38].

Typical bandwidth requirements depend on the size and application of the sub-network. Literature estimates vary between 100 kbps and 100 Mbps, and it is expected to cover ranges between 0.1-10 km [7, 36, 37]. A popular communication technology for NAN/FAN is WiMAX (IEEE 802.16), which supports wireless broadband up to 10 km. Other technologies include GSM, DSL, power line communication (PLC) and low earth orbit satellite systems [13].

Wide area networks: SG wide area networks (WANs) provide the backbone communication links and provides the long distant communication links between control centres and dispersed AMI systems. The typical bandwidth requirements for the SG WAN is around 10 Mbps to 1Gbps and is expected to cover ranges between 10-100 km [36]. Popular communication technologies for WAN includes WiMAX, optical fibre, LTE and geostationary satellite systems [13].

Comprehensive reviews of all of the above mentioned communication technologies for SG can be found in [13, 41, 42]. Some recent communication advances specifically for SG application is presented in [40]. These technologies include narrowband powerline communications (PLCs) like PRIME-PLC and G3-PLC, broadband powerline communications (BPL) and a wireless mesh-network standard, ZigBee.

2.1.2.2 SG communication middleware:

Several professional associations, international standardisation bodies, regional standardisation organisations and other industrial alliances have contributed significantly towards SG middleware standardisation. Although no single, exhaustive reference of SG middleware exist yet, the following specification documents are deemed by the authors of [8] as prominent: ANSI C12.1-2008, (American National Standards Institute) and IEEE 1377 (Institute of Electrical and Electronics Engineers) specify *syntax* between device and utility, using XML and binary documents. IEC 61968-9 (International Electro-technical Commission) specifies SG data management, prepaid metering and grid events. Reviews of SG standardisation documentation can be found in [13, 43, 44]. Surveys of popular middleware platforms are presented in [13, 45]

2.2 M2M standardisation efforts

This section presents a review of some of the recent research activities and technological developments in IoT and M2M technologies. As with SG, M2M is an umbrella term for a plethora of activities, making a full review of all recent efforts beyond the scope of this

work. A brief overview is presented after which the details of the specific technologies employed in this work are presented.

In the previous section it was found that the SG vision consists of two subsystems, a communication infrastructure, and middleware platform [13]. This idea is in fact not unique to SG but forms the point of departure of a general world-wide effort towards M2M standardisation as enabler for IoT.

2.2.1 Standard service layer for M2M

This section gives an overview of the recent world wide efforts towards developing a standardised service layer (i.e. middleware) for M2M communication. As the name suggests, the goal is to provided a standard interface with heterogeneous networks and devices through exposing common services.

2.2.1.1 History of M2M middleware standardisation

After conducting an in-depth study in 2008, in various application domains, the European Telecommunication Standards Institute (ETSI) listed a set of generic requirements for M2M communications. The effort resulted in technical specification documents [46, 47, 48] that were finalised in 2012. These specifications became known as the *SmartM2M* or *ETSI M2M* standards [49]. The specifications define a horizontal service platform called a *Service Capability Layer (SCL)*. The SCL provides open application interfaces that enables the development of M2M services and applications independent of the underlying network [50]. This layer therefore attempts to standardise common M2M challenges such a storage, addressing, discovery, access rights, etc., that get continually reinvented.

In 2013, ETSI (along with six other global telecommunication SDO's) formed the *oneM2M* alliance. The purpose of oneM2M is for the further development and maintenance of the SmartM2M specifications [51, 52]. The first official specifications of oneM2M was released in January 2015 [53]. Subsequently, oneM2M extends the SmartM2M standards by also investigating the communication infrastructure and standard semantics. A review of the oneM2M standards may be found in [51].

Substantial effort towards M2M standardisation has also been done by 3GPP with their MTC (Machine-Type-Communication) specification [54]. However, this specification focuses mainly on GSM access technologies like UMTS and LTE, and is currently been absorbed into the oneM2M specification.

2.2.1.2 SmartM2M Architecture

A high-level overview of the SmartM2M standards are presented in this section. The oneM2M architecture will not be discussed because no software implementations of this contemporary standard was available for public use at the time of the development of this work.

Figure 2.3 illustrates the high-level functional architecture of SmartM2M from [49, 55]. The figure shows a hierarchical structure that consists of three sub-domains, *device*, *gateway* and *network*. This architecture is similar to that of the SG communication architecture from Figure 2.2 on page 13, in that these subdomains corresponds roughly to HAN, NAN and WAN. (In fact, ETSI paid specific attention to standardisation of smart metering use cases and SG application in [56, 17].)

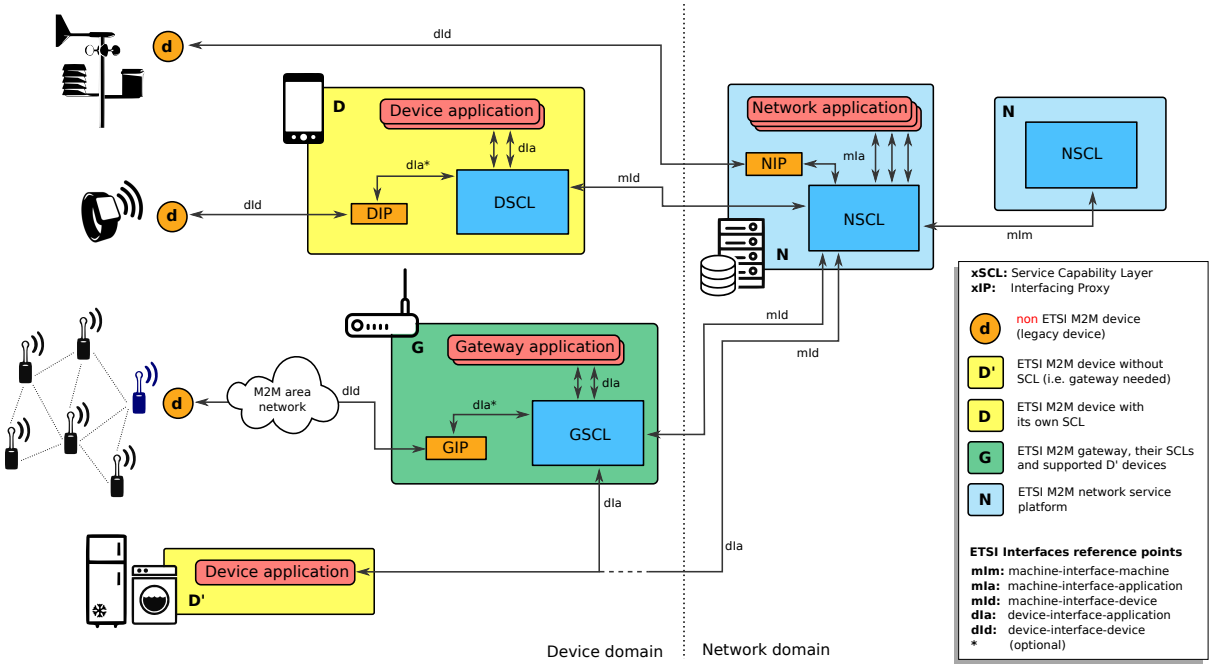


Figure 2.3: ETSI M2M high-level functional architecture with example devices. Adapted from [55, p. 35], [50].

xSCL: The standards specify what is called a *Service Capability Layer or SCL*, which is the application that forms the foundational building block of the middleware. The idea is that SCL instances run throughout the network in every domain. SCLs running at device level are referred to as **D**SCLs, in the core network as **N**SCLs and on gateways as **G**SCLs. Each SCL instance provides common services such as: machine registration, synchronous and asynchronous communication, resource discovery, access rights management, group broadcasts, *et cetera*.

Legacy devices: Understanding that adoption of the SmartM2M standards will happen incrementally, ETSI prescribed legacy device and network support as a requirement. This is implemented by what are called *interworking proxy units (IPUs)* that may be added as modules to an SCL. Again IPUs may reside at any level as shown in Figure 2.3 and are subsequently called *DIP*, *GIP* and *NIP*. As the name suggests, the purpose of an IPU is to appropriately translate the legacy protocols and mechanisms into REST requests for the residing SCL. The nature and complexity of any specific IPU depends on the translation.

Interfaces: Figure 2.3 also shows the defined interfaces *between* the separate entities *(mIm, mIa, dIa, dId)*. These definitions are used to specify the requirements of the different interfaces such as bandwidth requirements, network address translation (NAT) traversal capabilities, *et cetera*. It is important to note that the SmartM2M standards do *not* assume or enforce any specific network communication protocols. However, the architecture is designed to use currently popular protocols such as HTTP.

REST: SCL services are provided using a RESTful interface. The term REST originates from REpresentational State Transfer, which means:

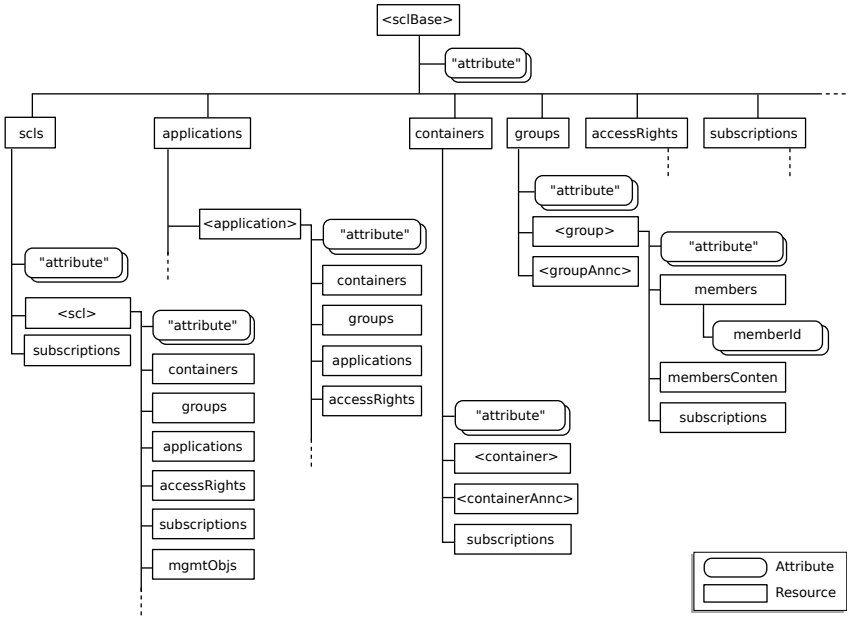


Figure 2.4: ETSI RESTful resource tree standardisation. Extracted from [55, p. 34]

- End-point nodes should be thought of as objects (called *resources*) instead of actions;
- Each resource has a universally unique ID (*URI*);
- Each resource remembers *only* its own state;
- State should be represented and transferred as web documents; and
- Access to the state of each resource is constrained to only four actions: *create, retrieve, update, delete* (CRUD).

Since no resource is responsible for remembering the state of any other resource than itself, the REST architecture is called *stateless*. Statelessness is important, since it enables horizontal scaling, that is, performance can be added by simply adding more machines. Statelessness also enables network caching which greatly aids resource contained devices by allowing data to be retrieved even when a device is off-line or limiting the amount of requests to an individual device. The statelessness and cacheable nature of the REST architecture has proved vastly scalable with the success of the web [57]. A highly scalable architecture is a key prerequisite for M2M. The inherent nature of M2M distributed systems is also more about *tangible states* than about *exposed methods* [55, p. 112] For these reasons, ETSI decided to adopt a REST architecture for the new standards.

Standardised resource tree: The resources in a REST architecture are conceptually arranged and addressed in a hierarchical manner. This is referred to as a *resource tree*. Each SCL in the SmartM2M network contains a resource tree through which services are presented. All interaction with the resource tree is done using only the CRUD operations defined by REST. The structure of this resource tree has been standardised by ETSI and is presented in Figure 2.4. The definition of the resources are as follows (from [47, p. 69]):

scls: Consists of a collection of all the *remote* SCL that *this* SCL is connected to. This may be other NSCLs, GSCLs or DSCLs.

applications: A collection of standard compliant applications that are registered with this SCL. These applications may store data at the SCL under the "container" resource of its branch to which other SCL clients may subscribe.

containers: A list of the containers created by local and remote entities for data storage.

groups: A list that defines logical groups of other resources and may be used to perform bulk actions, such as adding data or changing permissions.

accessRights: A list of resources that specify which entities may or may not access and modify specific resources at the SCL.

subscriptions: A list of resources that represent every individual subscription at the SCL. An external entity subscribes to a resource by providing the SCL with a RESTful endpoint of its own. This endpoint is referred to as a *Point of Contact (PoC)*. Whenever a change occurs on the resource in question, the SCL notifies the remote entity by POSTing a notification document to the supplied PoC. As shown in Figure 2.4, subscription may occur at every level.

2.2.1.3 Implementation

Shortly after the release of the SmartM2M standards in 2012, an open-source project initiated by LAAS-CNRS, called *OpenM2M* (also *OM2M*) started work on a reference implementation of the standards [58]. The result was a modular and extensible framework written in Java on top of the OSGi Equinox runtime. This includes the implementation of SCLs for both server and gateway devices [50, 59].

Similarly, development of the *OpenMTC* was started by Fraunhofer FOKUS and TU Berlin [60, 61, 62]. Contrary to OM2M, the OpenMTC platform is not open-source and unavailable for use in the public domain.

2.2.2 Semantics

At this point, it worth noting that the middleware interoperability layer can be divided into two domains (1) *communication* interoperability and (2) *semantic* interoperability as pointed out by [63]. The previous section only focused on communication interoperability. The semantic layer is the upper most layer of middleware. It specifies the *data models* and *business rules* of the application which essentially defines *meaning*. At the time of this writing, the SmartM2M standards only specify the former. This means that the *content* of the documents in the predefined RESTful resource tree is left unspecified and open to interpretation for each application.

Although in infancy, substantial effort has gone into developing *semantics* for M2M that provide universal data models to establish business level interoperability. Leading groups include the *IPSO alliance* [64] and the *Open Mobile Alliance Light Weight M2M (OMA LWM2M)* [65]. Most standardised semantics use XML or JSON to present the data models. However, the IPSO alliance also specify binary data structures to support resource constrained devices.

Application Protocols		DDS, **CoAP**, AMQP, **MQTT**, MQTT-SN, XMPP, HTTP, **REST**
Service Discovery		mDNS, DNS-SD
Infrastructure Protocols	Routing	RPL
	Network	**6LowPAN**, IPv4/IPv6
	Link	IEEE 802.15.4 (ZigBee)
	Physical	LTE-A, EPCglobal, IEEE 802.15.4, Z-Wave
Influential Protocols		IEEE 1888.3, IPSec, IEEE 1905

Table 2.1: Summary of M2M communication infrastructure that has received much attention recently [66].

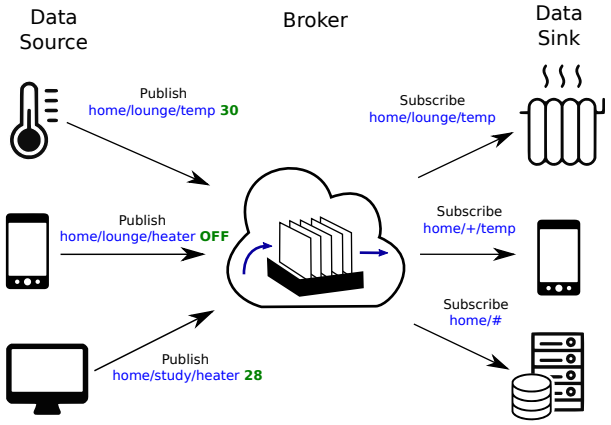


Figure 2.5: Example *Smart home* application using MQTT showing a many-to-many relationship via a broker.

2.2.3 Communication infrastructure technologies

The authors of [66] present a comprehensive survey on enabling technologies, protocols and applications for IoT. Table 2.1 is an excerpt from [66] and gives a summary of the separates network layers that have recently seen standardisation efforts. The main working groups involved with this standardisation is given as the: World Wide Web Consortium (W3C), Internet Engineering Task Force (IEFT), EPCglobal, IEEE and European Telecommunication Standards Institute (ETSI).

Due to design constraints (explained in Chapter 3), all layers shown in Table 2.1 except for the application layer have been fixed. In-depth attention will therefore only be given to the application layer to investigate design options.

2.2.3.1 Application protocols

Suitable application protocols allow for well separated business logic. Choosing an appropriate protocol is therefore a crucial step in the design process. There are a wide variety of application layer protocols for constrained environments. Each designed to address a specific domain of M2M or general communication requirements. Some of the most popular protocols used in M2M systems identified by [66, 67] are: AMQP [68], CoAP [69], DDS [70], HTTP [71, 72], MQTT [73], and XMPP[74]. Among these, MQTT and CoAP have received much attention recently and have been identified as good candidates worth investigating in this project. MQTT is popular for its large supporting communities and rich ecosystem of implementations. CoAP is renowned since it has been specifically designed for embedded REST.

MQTT: The *Message Queuing Telemetry Transport* protocol was introduced by IBM in 1999 and was standardised by OASIS in 2013 [75]. MQTT is a *publish-subscribe* protocol based on the concept of *topics*. Clients connect to a *broker*, which acts is the central server. A client may publish a message to the broker under a specific topic. This message is then pushed by the broker to any clients subscribed to that topic. Topics are arranged in a hierarchy separated by forward slashes ("/") as shown in Figure 2.5. Topic subscriptions may include *wildcards* in order to establish a one-to-one, one-to-many, or many-to-many relationship with data sources. It is important to note that the location of any client on the network is hidden from any other client, implying the only means of reachability is through abstract topics and not absolute addresses. This is contrary to most other application-layer protocols like CoAP or HTTP, which use a *Uniform Resource Locater* (URL) for addressing. A version of MQTT, called MQTT-SN [76] has also been released and is optimised for wireless sensor networks (WSN) [77].

MQTT also does not enforce any *semantics* within the topics or messages, placing this responsibility completely on the system designer. This freedom to structure the message as any character tuple is extremely powerful, but removes any common interfaces for cross system interoperability. As evidence of the lightweight, scalable, and efficient nature of MQTT, the popular social media site, Facebook, announced in 2011 that their chat application, which consists of millions of users world-wide, uses MQTT [78].

CoAP: The *Constrained Application Protocol* was introduced by the Internet Engineering Task Force (IETF) in 2013 [69]. The motivation for its creation is to introduce the REST architecture to constrained environments [79]. CoAP is therefore different from the other M2M protocols mentioned in that it has both a request-response and publish-subscribe architecture, as opposed to only a publish-subscribe architecture. For this reason, CoAP can interface *directly* with the SmartM2M and oneM2M SCL described in the previous section.

HTTP is a well known REST-based application protocol, and has proved to be one of the core actors for the vast scalability of the Web [57]. However, HTTP lacks some features needed for M2M applications, for example *asynchronous* messages (publish-subscribe), low bandwidth headers and lossy transport tolerance. These are some of the challenges addressed by CoAP.

In essence, CoAP was designed as a lightweight, binary version of HTTP but with the following key differences:

- The *GET* method was extended to include an *observe* functionality, which allows a publish-subscribe mechanism alongside the request-response architecture.
- Binary headers are employed at a fixed width of 4 bytes for a low bandwidth footprint.
- CoAP was designed to be transported over the *User Datagram Protocol (UDP)* which is more suitable for lossy wireless sensor networks (WSN). However, UDP does not guarantee delivery or allow NAT traversal (being connectionless). CoAP addresses this by allowing *confirmable* messages and *piggy-back* responses.
- CoAP has inherent support for the *6LoWPAN* [80] protocol (IPv6 over Low power Wireless Personal Area Networks), making it state-of-the-art.

	MQTT	CoAP
Standards	OASIS	IETF
Est.	1999	2013
Server implementations	Mosquitto, HiveMQ, Mosca, IBM Bluemix	Californium (Java), CoAP.NET(C#), gen_coap (Erlang), node-coap (JavaScript) [86]
Client implementations	Paho (C/C++/C#, Python, JavaScript, Java, Android) [87]	libcoap (C), SMCP (C), Wakaama (C) [86]
Header	2 bytes	4 bytes
Transport	TCP	UDP
QoS levels	3	2
Architecture	Pub/Sub	Req/Resp (REST) and Pub/Sub
Security	SSL/TLS	DTLS
M2M semantics	None	IPSO, LWM2M
Automatic discovery	No	Yes
Community support	Rich	Poor
Lightweight	Yes	Yes

Table 2.2: Comparing MQTT and CoAP

Another aspect of the REST architecture of CoAP is that is has a more rigorously defined structure compared to most other application layer protocols. The well defined structure of REST lends itself towards building standard data models for M2M *semantics*. For this reason, organisations like the IPSO Alliance [64] and OMA [81] are specifically developing M2M semantic standards assuming CoAP as the underlying application protocol. CoAP is also optimised for the 6LoWPAN (discussed next) network protocol designed for lossy wireless sensor networks such as ZigBee. For these reasons CoAP is seen as fundamental to M2M. Since CoAP is a relatively new protocol, the ecosystem for open-source, community-driven implementations of CoAP and related tools are still relatively underdeveloped.

Finally, since MQTT is popular lightweight protocol for M2M, the more recent CoAP protocol stands in direct competition. For this reason comparisons of MQTT and CoAP is frequently found in literature [82, 83, 84, 85]. Table 2.2 summarises the main differences.

2.2.3.2 Network protocols

IPv6 over Low power Wireless Personal Area Networks or 6LoWPAN was developed by IETF which, released its specification in 2007 [80, 88]. As the name suggest, the purpose of 6LoWPAN is to introduce the IP protocol to lossy wireless networks that consists of resource constrained devices. 6LoWPAN is generally envisioned as the network layer protocol in Zigbee radio mesh networks, with CoAP as the application layer protocol [79, 89]. A major challenge is that IP software stacks are relatively heavy operations and not suitable for constraint devices. Much effort has gone into developing lightweight IP stacks for 8-bit environments. Specifically the *Contiki* project [90] delivers an operating system which implements the 6LoWPAN stack as well as CoAP for embedded systems [91, 92]. Other embedded IP implementations include lwIP [93], TinyOS and MagnetOS [91].

2.2.4 Summary of M2M standards

The following key points on M2M standardisation are surmised. New middleware specifications for M2M, based on the REST architecture represent a significant step towards large-scale M2M interoperability. However, the contemporary nature of the specifications mean that they have not yet been widely adopted and therefore lacks implementation.

M2M efforts in communication infrastructure have seen development at all network layers. Specifically at the application layer, CoAP promises seamless integration with both the above REST middleware architecture and lower M2M layers like 6LoWPAN and Zigbee. However, CoAP also suffers from lack of adoption due to its recentness and is currently competing with MQTT, a widely adopted and supported lightweight publish-subscribe protocol.

2.3 EWH in the Smart Grid

This section discusses the suitability of targeting EWHs for DSM. Subsequently, the thermal modelling of EWHs are presented in order to do so effectively. Since this area of the work is chiefly concerned with validating the models presented by Nel [26], the focus falls on the description of these models.

EWHs along with HVAC are classified as *controllable thermostatically controlled appliances (C-TCAs)* meaning that they can act as an energy buffer to store energy in the form of heat for delayed use. According to [94, 95, 96] this characteristic makes EWHs suitable for power scheduling schemes. However, in order to do this the scheduling scheme needs to account for the device thermal behaviour, random consumption and customer comfort. Furthermore, the authors of [27] estimates an energy reduction between 14% and 17% is possible by implementing scheduled heating control. This means that scheduling EWH may not only be used to for load-shifting but also for overall load reduction. Doing so, however, also requires EWH modelling to ensure customer satisfaction is observed.

2.3.1 EWH modelling techniques

The approaches for modelling may broadly be classified into two categories [96]: (1) aggregate load control for a large number of EWHs, (2) physical models for individual EWHs. Aggregate models are typically more suited for improving overall electrical network efficiency and stability, but cannot guarantee individual customer satisfaction. Physical models have the advantage of taking user comfort into account but at the cost more computational resources required for modelling each individual EWH. Since customer satisfaction is a key requirement for this investigation, aggregate models are not investigated.

Three general approaches to modelling individual EWHs are found in literature namely, (1) data-driven, (2) physics-based and (3) grey-box [97]. The data-driven approach uses data collected of the system during normal operation and attempts to find a relationship between the input and output variables using statistical techniques. It is therefore also known as the inverse or black-box approach. In the second approach, the model is derived by using the underlying physical laws that govern the process. The physics approach is also known as the forward or white-box approach. Finally the third approach, is to use a combination of the two former approaches; the basic operation of the model is defined using physical laws, but the parameters for the model are estimated using measured data on the system. Both methods (2) and (3) require in-depth knowledge of the system [97].

An EWH is classified as a low-pressure (between 100-600 kpa), stratified thermal energy storage system (STESS). Several techniques of varying degree of complexity exists for modelling STESS systems, an overview of which can be found in [98]. However, when investigating individual EWH modelling for the purposes of DSM, it is important that the

computational complexity is low to enable implementation scalability. This requirement narrows the types of physics-based models down to what can be called *nodal* models. Nodal models simplify things by assuming the internal water to be a static mass (or multiple static masses) and do not take into account any dynamic effect such as fluid turbulence, velocity, viscosity, *et cetera*.

2.3.2 EWH nodal models used in SG applications

Nodal models for EWHs are used extensively in literature for SG application. For example, in [94] an appliance commitment algorithm was developed using an EWH model to optimally schedule heating according to price-forecasting. The aim of the algorithm is to find the optimal user comfort for minimum payment. A one-node EWH model is used with a random hot water usage profile. However, the algorithm was tested *only* in simulation and has not been validated for accuracy against experimental data.

The authors of [27] propose a proof-of-concept large-scale EWH monitor and control architecture. They use a one-node model to illustrate the benefit of EWH scheduling. Two shortcomings are identified here: Firstly, model parameters are *assumed* and not measured in a controlled environment. Secondly, the model results are not validated using laboratory experiments.

Similar examples of nodal models for SG application may be found in [33, 95, 99, 100]. Again, these propose nodal models for EWH prediction in SG application. However, accuracy of these models do no appear to have been validated by rigorous laboratory experiments.

2.3.3 Thermal principles of nodal models

Before moving on to discuss the details of the nodal models by Nel [26], a brief overview of the underlying heat and mass transfer principles governing EWH behaviour is summarised below.

Water enthalpy: Water *enthalpy* refers to the *heat energy* contained in a body of water as a result of its *temperature* [101, p.8]. The enthalpy of a single volume of water, assuming a uniform temperature can be calculated as follows:

$$\begin{aligned} U_{\text{cyl}} &= cm_{\text{cyl}}T_{\text{cyl}} \\ &= c\rho V_{\text{cyl}}T_{\text{cyl}} \end{aligned} \tag{2.1}$$

Where U_{cyl} is the enthalpy contained in the EWH cylinder in *Joule*, c is the specific heat constant for water, 4184 (J/kg.°C), m_{cyl} is the mass of the water in kg, ρ the density of water at 1000 kg/m^3, V_{cyl} the volume of water in m^3 and T_{cyl} is the average temperature of the water in the EWH cylinder in °C.

Enthalpy transfer: The enthalpy contained in a EWH cylinder, U_{cyl}, may be *decreased* by means of two possible mechanisms: (1) water is drawn-off at the outlet for usage and is replaced with colder water entering at the inlet, and (2) heat is radiated to the environment through the surface of the EWH cylinder. The EWH enthalpy may be *increased* by means of the electric heating element. Each of these energy transfer mechanisms are discussed below.

Draw-off energy: The energy leaving and EWH cylinder due to warm water draw-off is calculated as follows:

$$Q_{\text{draw}} = c\rho V_{\text{draw}}(T_{\text{cyl}} - T_{\text{in}}) \tag{2.2}$$

Where T_{cyl} is the temperature of the water leaving the cylinder, T_{in} is the temperature of the water entering the cylinder and V_{draw} is volume of the water drawn-off. In this form the inlet temperature T_{in} is used as the reference point for **zero** enthalpy.

Thermal radiation: Thermal radiation occurs across a temperature boundary between two masses and causes the warmer mass to cool down and the cooler mass to warm up until a state of equilibrium is reached. For an EWH the thermal radiation is important since it represents energy losses to the environment. Thermal radiation occurs according to Newton's exponential law of cooling, which states that the rate of thermal radiation is proportional to the difference in boundary temperature. It is expressed as a first order differential equation (DE) as follows:

$$\frac{\delta}{\delta t}T_{\text{cyl}}(t) = -k(T_{\text{cyl}}(t) - T_{\text{amb}}(t)) \tag{2.3}$$

where T_{cyl} is the water temperature of the EWH cylinder, T_{amb} is ambient temperature and k is the cooling constant, which is dependent on the boundary insulation material of the EWH. Equation 2.3 can be put in terms *heat energy* transfer by substituting T_{cyl} from Equation 2.1:

$$\begin{aligned} \frac{1}{c\rho V_{\text{cyl}}}\frac{\delta}{\delta t}U_{\text{cyl}}(t) &= -k(T_{\text{cyl}}(t) - T_{\text{amb}}(t)) \\ \frac{\delta}{\delta t}U_{\text{cyl}}(t) &= -c\rho V_{\text{cyl}}k(T_{\text{cyl}}(t) - T_{\text{amb}}(t)) \\ &= \dot{Q}_{\text{loss}}(t) \end{aligned} \tag{2.4}$$

Where, $\dot{Q}_{\text{loss}}(t)$ is the rate at which thermal energy is lost due to radiation, in *Watt*. From this a thermal resistance constant of the EWH cylinder may be defined as:

$$R \triangleq \frac{1}{c\rho V_{\text{cyl}}k} \tag{2.5}$$

where R is measured in °C/Watt and V_{cyl} is the volume of the cylinder. The thermal resistance constant represents the ability of the EWH cylinder to retain its thermal energy. The higher the resistance, the better an EWH is at retaining its enthalpy.

Input energy: Energy is added to an EWH cylinder by electrical heating. This is expressed simply as:

$$\dot{Q}_{\text{elec}}(t) = P_{\text{elec}}(t) \tag{2.6}$$

Feedback model: The above equations may be expressed as a feedback system acting on a single volume EWH. This is depicted in Figure 2.6. The model also shows the operation of the thermostat.

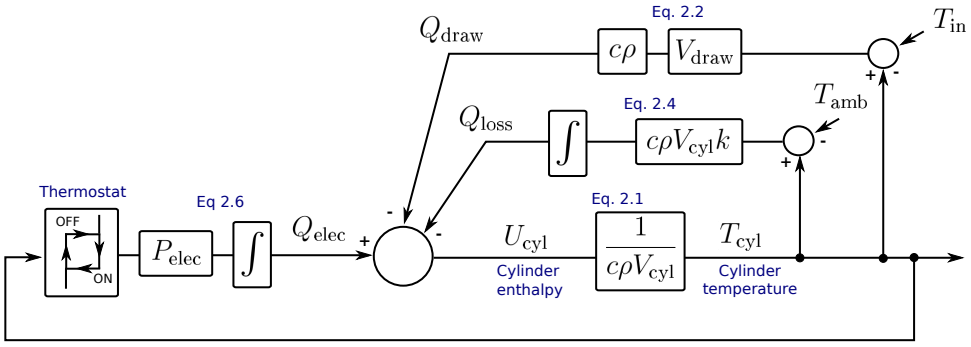


Figure 2.6: Summary of the thermal transfer mechanisms for a single volume EWH cylinder including thermostat feedback.

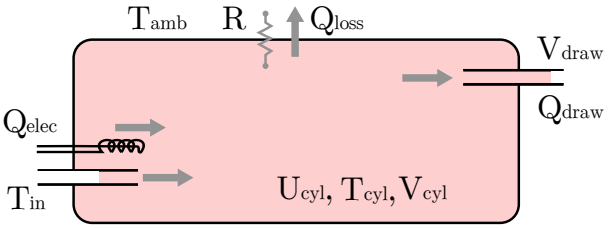


Figure 2.7: One-node model energy transfer representation.

2.3.4 Energy model by Nel

In preparation for their validation, an overview of the models proposed by Nel [26] is now presented. Nel proposes two models which are based on the *nodal* approach for computational efficiency. The first model is a *one-node* model, which treats the internal body of water as a single uniform mass. The second model is a *two-node* model and is derived from the simpler one-node model and treats the internal body of water as two uniform masses. The two-node model is used to model the *stratification* phenomenon typically present in hot water cylinders where lower-density warm water tend to rise above the colder water and create a *thermocline* that separates the two masses. Both models are based on the energy balance assumption that was shown in Figure 2.6. Energy is only put into the system using the electric element. Energy only leaves the system either through water draw-off events, or through thermal losses. Note that the equations governing the behaviour for the models are presented as *discrete functions* so as to indicate how they are implemented in simulation.

2.3.4.1 One-Node model

Figure 2.7 depicts the one-node model. All the water in EWH cylinder is treated as a *single* body with uniform temperature. Therefore, when a draw-off event occurs, the water leaving the cylinder through the outlet pipe is assumed to be at the average temperature of the water inside the cylinder (T_{cyl}), and the heat energy (enthalpy) leaving the system is referred to as Q_{draw}. Additionally, the cold water entering the cylinder from the inlet pipe (to replace the used water) is assumed to instantaneously mix with the water inside the cylinder at the start of the present sample, to create a new average temperature. Standing losses (Q_{loss}) occur according to Equation 2.4 and is dependent on some thermal resistant constant, R. Heat is added to the single node via the electric element and referred to as Q_{elec}.

Usage: The delta enthalpy leaving the system due to a draw-off is calculated using Equation 2.2. Given that the draw-off volume and inlet and outlet temperatures are known, Equation 2.2 can be rewritten as a discrete function:

$$Q_{\text{draw}}[n] = \rho\, V_{\text{draw}}[n]\Big(T_{\text{cyl}}[n] - T_{\text{in}}[n]\Big) \tag{2.7}$$

Where c is the specific heat capacity of water, ρ is the density of water, $T_{out}[n]$ is the temperature of water leaving the EWH at sample $[n]$, and T_{in} is the temperature of the water entering the cylinder from the inlet. A usage event, e.g. running a bath, will span many samples. Energy lost through the entire event can be calculated by summing the energy losses over the sample periods.

Element heating: For the one-node model, energy input by the element (Q_{elec}) is assumed to be distributed uniformly and instantaneously to all the water in the EWH cylinder. $Q_{\text{elec}}[n]$ is derived from the power rating of the element and the time it is estimated to be over the sampling interval.

The temperature *increase* of the water in the cylinder as a result of energy input by the element can be calculated rewriting Equation 2.1 as a discrete function as:

$$T_{\text{cyl}}[n+1] = T_{\text{cyl}}[n] + \frac{Q_{\text{elec}}[n]}{c\rho V_{\text{cyl}}} \tag{2.8}$$

where, from Equation 2.6:

$$Q_{\text{elec}}[n] = P_{\text{elec}}[n]\Delta n \tag{2.9}$$

Standing losses: Standing losses refer to the energy lost due to heat dissipation from the single water mass inside the EWH to the outside environment as a result of the temperature differential. These standing losses are modelled using a temperature decay of the water inside the EWH toward the ambient temperature as described by Equation 2.3. Rewriting Equation 2.3 as a discrete function, and substituting k for R using Equation 2.5, the change in internal temperature due to thermal radiation can be calculated as follows:

$$T_{\text{cyl}}[n+1] = T_{\text{cyl}}[n] - \frac{T_{\text{cyl}}[n] - T_{\text{amb}}[n]}{c_p\rho V_{\text{cyl}} R} \tag{2.10}$$

Where R is the thermal resistance of the EWH cylinder and T_{amb} is the temperature of the outside environment of the EWH.

The energy lost to the environment over a sampling interval, Q_{loss}, can then be calculated using:

$$Q_{\text{loss}}[n] = c\,\rho\, V_{\text{cyl}}\Big(T_{\text{cyl}}[n] - T_{\text{cyl}}[n+1]\Big) \tag{2.11}$$

2.3.4.2 Two-Node model

The two-node model is an extension of the one-node model, for which the cylinder is split into two one-node nodes, each with its own temperature, volume, and standing losses. A

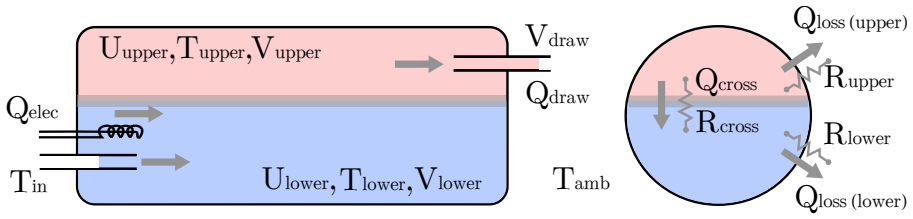


Figure 2.8: Energy flow, upper and lower nodes of two-node state.

depiction of the two node model is shown in Figure 2.8. Notice that with the two-node model, Nel [26] introduces a fourth energy exchange mechanism which is the *inter-nodal* energy transfer, Q_{cross}, where the warmer, upper node radiates heat to the cooler, lower node across the thermocline. This mechanism attempts to model the gradual mixing of the two nodes.

An important aspect of the two-node model, which is difficult to depict in the figure is that the model *transitions* between one-node and two-node state (and vice-versa), depending on model activity. The operation of the two states can be described as follows:

One-node: The modelled EWH remains in the one-node state until a significant volume of water, called the threshold volume (V_{thres}) is consumed, after which the model transitions to the two-node state, mimicking the natural stratification that occurs in the EWH. Afterwards, the upper node consists of the remaining warm water in the cylinder after a usage event has occurred and the lower node consists of cold water from the inlet that has replaced the water drawn from the cylinder.

Two-node: In the two-node state, usage leads to water drawn from the upper node, which is then replaced by water entering the lower node through the inlet pipe, effectively raising the thermocline. It is assumed that the inlet temperature mixes instantaneously only with the lower node. Furthermore, all energy put in by the EWH element is also assumed to be transferred to the lower node *alone*. The modelled EWH remains in the two-node state until one of three events occur: (1) enough draw-off events occur such that $V_{\text{up}} <= V_{\text{thres}}$, (2) the lower node is heated by the element until $T_{\text{up}} = T_{\text{low}}$ or (3) enough time passes such that cross thermal radiation causes $T_{\text{up}} = T_{\text{low}}$. The heat and mass transfers of the two-node model is modelled as essentially two one-node models, with varying volumes and thermal resistances according to Equations 2.7 to 2.11. More details can be found in [26, 102].

EWH orientation: For the two-node model, it is important to take the *orientation* of the EWH into account, since the geometry of the internal and external surface areas of the nodes are affected. For a *vertical* EWH, the thermocline area stays constant as it moves along the height, but varies for that of a *horizontal* EWH. Similarly, the external surface of a vertical EWH varies longitudinally, whereas for a horizontal EWH it varies along the latitude. A contribution by Nel [26] was to develop a computationally efficient way of calculating the arc length of the circular segment of a horizontal cylinder given the radial distance (which is a trigonometric problem with no closed-form solution) using the Newton-Raphson method. This method is also detailed in [102]. By this method both the cross-sectional area of the thermocline, as well as the exposed external surface area of each node can be calculated. The thermal resistance used to calculate the standing losses

	Mk2	Mk3
Computation	Particle core (ARM M3, 32-bit, 72MHz)	Atmel Xmega128A4U (16-bit, 38 MHz)
Communication	2G modem: Maestro EVO 100 (Transparent UDP mode)	2G modem: Ublox G100 (Opaque TCP mode)
Sensors	4 Temperature sensors 2 Flow meters Indutive power meter	
Actuators	Latching Relay Latching valve	

Table 2.3: Summary of features of Mk2 and Mk3 hardware by Brown [25].

and thermal decay of a node, is dependent on the surface area of the node exposed to the environment. The exposed surface area for a horizontal EWH consists of (1) the area of the circular segments on either side of the cylinder, which are identical; and (2) the area of the rectangle that makes up the portion of EWH wall for a particular node. For a vertical EWH cylinder the cross-sectional area remains constant, eliminating the need for the estimation of arc length. This reduces the complexity of the calculations as the surface areas of the two nodes can be calculated analytically to determine the standing losses for each node separately.

2.3.5 Summary of EWH modelling

EWHs have been motivated as good target candidates for load shifting and reduction DSM schemes. This requires modelling the EWH to ensure customer satisfaction. Different modelling approaches exist, however the *nodal* approach is the most computationally efficient and therefore appropriate for large scale application. In fact some EWH DSM schemes have been proposed in literature using nodal models, however, it was found that the accuracy of these models have not been validated rigorously. The heat- and mass transfer principles of nodal models, followed by an overview of the one-node and two-node models proposed by Nel [26] are presented.

2.4 Sensory hardware by Brown

As stated in the Introduction (Objective 1b), this study employs the hardware developed by Brown [25] as the measuring and control equipment for EWHs. As part of Objective 1, a communication stack needs to be implemented on the hardware that is in line with the SG and M2M vision. This hardware is used since redeveloping new hardware falls outside the scope of this work. However, using pre-existing hardware introduces constraints on the range of design decisions. The purpose of this section is to describe the capabilities and features of the hardware.

Three EWH controllers are described in [25], two of which are applicable to this work. In keeping with the naming convention used in [25], the two controllers will be referred to as **Mk2** and **Mk3**. Table 2.3 summarises the features of the Mk2 and Mk3 controllers. Photos of the hardware may be found in Appendix A. A discussion follows below.

2.4.1 Sensors

Temperature sensors: Analogue temperature sensors are used to measure four areas of interest around the EWH. All temperature measurements are performed *externally* (i.e. non-invasively), which makes it easier and safer to install. However, this trades off accuracy since internal water temperature cannot be measured. The temperature sensors are accurate to within 1°C assuming calibration. The measurements are as follows:

1. **Outlet:** Directly at the outlet, on the pipe surface.
2. **Far:** 500 mm along from the outlet, on the pipe surface.
3. **Inlet:** 500 mm along the inlet, on the pipe surface
4. **Ambient:** In the air, within 1 m radius from the EWH.

Flow meters: An in-line flow-meter at the inlet allows for the measurement of usage event sizes. The flow meters measures in 0.5 litre increments by using a Hall effect sensor.

Drip detection: Drip detection is performed by measuring the resistance between the two forks of a probe placed in the drip-tray. Drip detection is not used in this study.

Power meter: An inductive loop (current transformer) is used to measure the supply current to the EWH. The supply voltage however is not measured and assumed to be 230 Volts RMS. The power calculated is using the standard AC power formula $P = V_{RMS} \times I_{RMS}$. The accuracy of the sensor was verified to be within 5%.

2.4.2 Actuators

Latching devices were chosen by Brown to ensure that power failure does not alter the state of the valve or the relay.

Latching relay: To switch the element of the EWH, a latching relay is used. The original mechanical thermostat is **not** replaced but simply placed in series with the latching relay. The mechanical thermostat, being set to its highest temperature of around 70°C, acts as an added safety precaution.

Latching valve: A valve at the inlet of the EWH allows remote shut-off of supply water in the event of a burst detected by the drip detector. This is not used in this work.

2.4.3 Computation

For the purposes of this study, the main difference between the hardware for **System A** and **B** is in the microcontrollers and the 2G modems.

Mk2 - Particle core: This is a recent development kit from *Particle* [103]. It is an Arduino derivative hardware environment that is programmed using a web interface along with supplied software libraries. It is designed for high level rapid prototyping.

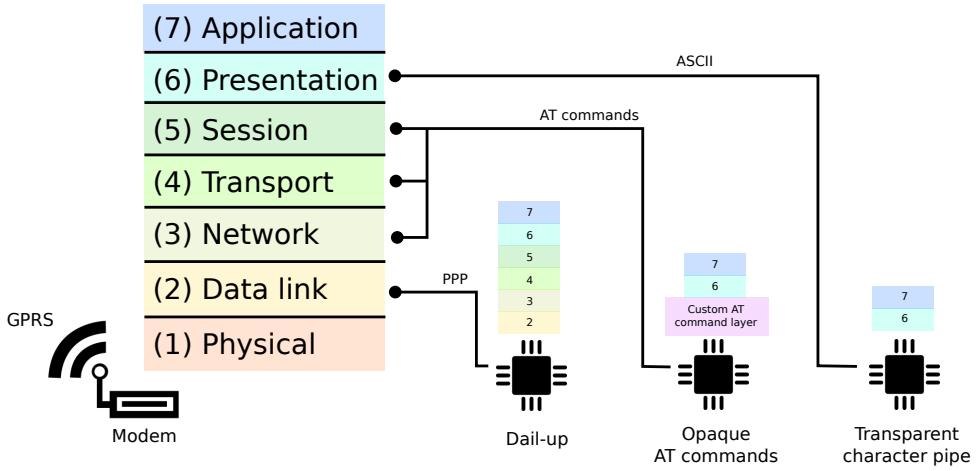


Figure 2.9: OSI stack representation depicting three methods of using a cellular modem for GPRS services.

Mk3 - Atmel Xmega: 8-bit microcontroller that is programmed in standard C [104].

2.4.4 Communication

Both Mk2 and Mk3 EWH controllers are equipped with 2G modems. The modems transacts all communication to the cellular network and presents an *application interface* through serial communication to the microcontroller. Most cellular modems still make use of the *Hayes AT command set* as application interface; a legacy modem command set introduced in 1981 [105]. A brief overview of the AT command interface is presented below since it forms a key point in this work.

2.4.4.1 Basic AT commands

The *"AT"* is an abbreviation for *attention* and always forms the first two characters of any command. For example, to dial a number, the following AT command would be used `ATD0826222538`. To query the current cellular operator, the command `AT+COPS?` is used. All AT commands are executed using a *carriage return* character.

2.4.4.2 Packet-switch services using AT commands

With the evolution of the cellular network over time – specifically the addition of *general packet radio service* (GPRS) – the AT command set grew to accommodate the additional functionality. This became known as the extended AT command set. However, these additional AT commands were never standardised and are still vendor specific.

In this study, cellular modems are used to connect to the Internet using GPRS. This is known as a *packet-switched* connection since data is carried in the payload of IP packets that are *routed* to their destination using an IP address. This is in contrast to the more traditional *circuit-switch* network, where an end-to-end circuit connection is set up by the network provider for voice or SMS. To establish a data session (known as a *packet-data-protocol (PDP) session*), the AT command set may be used in **three** different modes: (1) dial-up mode, (2) opaque command mode, or (3) transparent pipe mode. The latter two methodologies were added to the original Hayes AT command set and is now part of the *extended* AT command set. The modes are distinguished by the network layer at

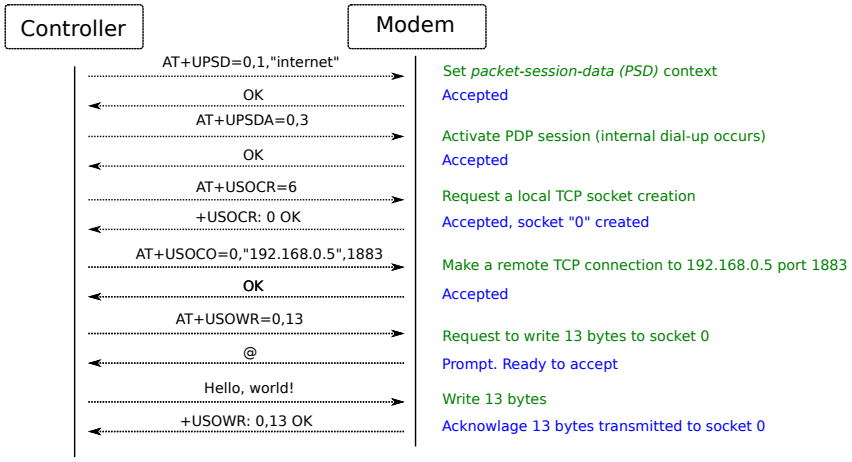


Figure 2.10: Example TCP transmission using opaque AT command mode

which communication between the modem and the microcontroller occurs. Figure 2.9 shows level at which each mode interacts. The importance of this distinction will become apparent in the discussion below.

Dial-up: This is the traditional method for establishing a PDP session. When a specific code is dialled using the AT command, `ATD****9*#`, the base station interprets that call as a *packet session* (as opposed to a *circuit-switched* session) and routes the connection to the core IP network. On the device, the modem switches the serial connection from AT command mode to the *Point-to-point protocol (PPP)*, a link layer protocol to carry the packet data frames [106]. Importantly all IP (network layer) related processing must occur on the connected microcontroller. Therefore, the microcontroller needs to have an IP software stack that can interface with the PPP connection. As discussed in Section 2.2.3.2, this requirement demands significantly more processing capabilities and software complexity. For this reason many modem vendors extended the AT command set to include two more ways of performing IP communication without the need for an IP stack on the microcontroller.

Opaque AT command mode: In this mode IP related actions are performed using AT commands only. In this case the IP software stack resides on the modem and is invisible to the controller. The AT commands therefore packages layers 3, 4 and 5 as a few select commands. Figure 2.10 illustrates the procedure to setup and transmit a simple byte message over TCP using opaque mode. This mode is discussed in depth in Section 3.3 on page 41.

Transparent character pipe mode: This mode has the largest abstraction gap between the Network and Application layer. In this mode, the modem is pre-configured with a host IP address and port. On start-up the modem automatically sets up a transparent end-to-end duplex character stream to the remote host. The onus falls on the modem to negotiate and manage the underlying IP packets and transport layer protocols. Little flexibility is left on the side of the controller. A major drawback is that in most cases only ASCII characters are allowed. This eliminates the possibility of using a protocol with binary control bytes (crucial for light-weight) on the application layer. The benefit

is that no effort is required from the connected microcontroller and can consequently be very simple.

2.5 Conclusion

Developing a EWH application for the SG requires a review of three domains: (1) the envisioned future SG architecture, (2) M2M communication standards for both middleware and communication infrastructure, and (3) thermal modelling of EWHs. From the review, several conclusions are as follows.

The future SG architecture is vast and convoluted with a plethora of standards, protocols, communication technologies and applications. Although no single exhaustive reference architecture for SG exists, standards such as the NIST SG framework and IEEE 2030-2011 are considered prominent stepping stones to this end.

M2M standardisation has enjoyed much attention recently by seeing the release of the *SmartM2M*, and more recent, *oneM2M* standards, which specify a full end-to-end middleware architecture based on REST. However these standards are new and therefore not yet widely adopted. On the other hand, M2M protocols at the application layer (such as MQTT and CoAP) and at the network layer (such as 6LoWPAN) are found to be more mature and more widely adopted.

Nodal models for EWHs are often found in literature at the core of DSM load shifting scheme propositions. However, little information exists on the accuracy and validity of these models and are often simply assumed to yield adequate results.

This study aims to contribute to the above literature in three ways: (1) provide insight and appreciation of common SG challenges by developing an SG application, (2) serve as a reference example with which to grasp the specifications by employing M2M standards, and (3) provide comprehension of model behaviour, accuracy, strengths and weaknesses by subjecting the nodal EWH models to rigorous experimental conditions.

Chapter 3

End-to-end M2M system design

This chapter presents the design and implementation of the M2M communications systems. In Section 1.5 it was stated that two M2M systems were designed, each investigating a different aspect of M2M. The focus of the initial design (System A) is standard compliance. More specifically the aim was to evaluate and attempt to satisfy the *high-level interoperability* requirement of IoT since this is the popular focus of current IoT research and development. This is done by implementing the EWH M2M system according to the SmartM2M standards. Since this system was only intended to be a proof-of-concept prototype serving 10 EWH clients, scalability and efficiency was given a lower priority. These requirements are the point of focus for the second design (System B), which was designed for 450 EWH clients. This is done by implementing the MQTT lightweight application layer protocol along with popular web technologies. In a synopsis: System A focuses on the *middleware layer* of SG, whereas System B focuses on the *communication infrastructure* of SG [13].

3.1 System A: Standard compliant middleware

3.1.1 Requirements and constraints

To conform to the Smart Grid and M2M/IoT vision of a distributed, decentralised architecture (Objective 1), it was decided to follow the SmartM2M specification. At the time of this writing, the only open-source implementation of the SmartM2M standards was the OM2M platform [50, 58]. OM2M was subsequently used as the middleware platform. Using the EWH hardware controllers by Brown [25], imposes two restrictions concerning the SmartM2M standards. Firstly, the *Particle* microcontroller (see Section 2.4.3) cannot support the OM2M implementation of the SCL and is therefore classified as a legacy device (see Section 2.2.1.2). Secondly, since the modem connects directly the cellular access network, the *gateway* concept of SmartM2M is rendered obsolete. This implied that only the NSCL component of the SmartM2M standard could be used.

3.1.2 System layout

The layout of System A is depicted in Figure 3.1. It should be cross-referenced with the ETSI standards architecture of Figure 2.3 (page 16). Below follows a discussion of the layout, the definition of the individual components, a discussion on the interaction between the components, and finally a brief discussion of the practical implementation.

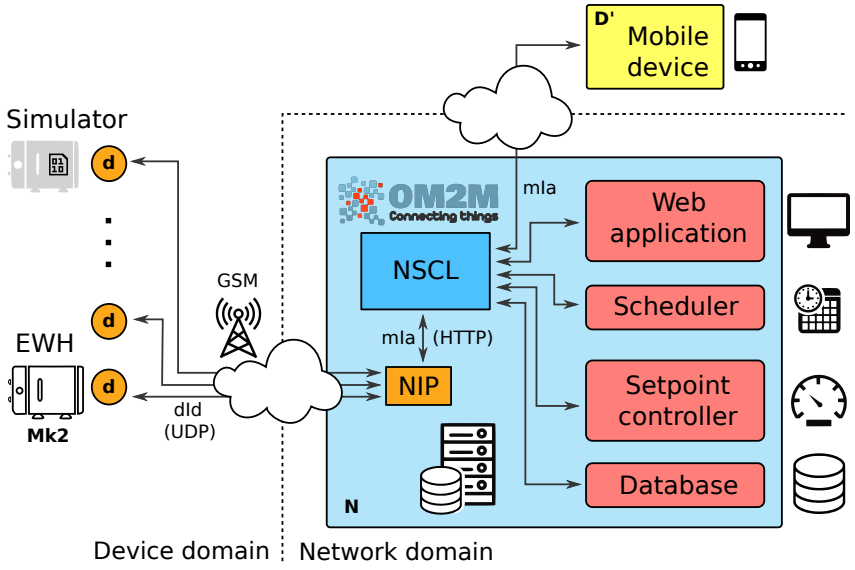


Figure 3.1: Architecture of System A implementing the standardised architecture of ETSI M2M (cross-reference with Figure 2.3 on page 16).

NSCL: The NSCL is the central point of interaction for System A. Its main purpose is to provide a request-response and publish-subscribe service between the individual components of the system. This is the Java implementation of NSCL that ships as part of the *OM2M* platform, as discussed in Section 2.2.1, and complies with the SmartM2M standards. The RESTful resource architecture of the NSCL and component interaction are discussed in detail later in Section 3.1.3 after the roles of all the individual components of System A have been defined. Since a gateway architecture is not applicable to the hardware (due to the cellular interface), only the NSCL component was used, omitting the GSCL altogether.

EWH client: The hardware constitutes the **Mk2** EWH controllers developed by Brown [25] as discussed in Section 2.4 (page 28). The microcontroller of the Mk2 hardware is a proprietary prototyping platform (*Particle core*) that has limited software libraries. This constrains the modem interface so that only the *transparent character pipe* AT command mode may be used to establish an Internet connection. This is an important constraint since it does not allow the use of binary characters that are found in most lightweight application protocols for M2M such as CoAP and MQTT. In addition to the EWH hardware, a simulator was implemented to allowed flexible development and testing of the end-to-end system.

NIP (Network Inter-working Proxy): The NIP is a custom built, server-side proxy that performs bidirectional translation of requests between the EWH clients and NSCL. The NIP is a necessary middle-man since at the time of development the onboard microcontroller did not support either the HTTP or CoAP protocol that would allow *direct* interaction with the NSCL. In terms of the SmartM2M standards, the EWH device is classified as a *legacy* device and requires either an NIP or GIP to interact with the architecture. This design is therefore fully in line with the SmartM2M vision albeit not the ideal case.

The NIP employs a open-source Java API to interface with the NSCL [107]. The API enables appropriate REST requests to the NSCL and automatically parses the XML

payload of each request into Java objects.

Web application: This is a simple web application to monitor and control the EWH Mk2 units for administrative purposes. The page is hosted with an Apache2 server [108]. Website actions translate to HTTP request to the NSCL to collect data or control EWHs. A screen capture of the web application can be found in Appendix B.

Mobile application: This is an application developed by Nel [26] for users to monitor and control their EWH. It was adapted to interface with the NSCL using the same API requests as the web application above. However, this was only done as a proof-of-concept and not deployed for use.

Setpoint-controller and Scheduler: The *setpoint-controller* is responsible for monitoring the temperature of each EWH registered on the system, and to issue relay state commands to the EWHs that do no satisfy the the required temperature. The *scheduler* maintains a *list* of setpoints which make up a 24 hour temperature schedule. The scheduler provides these setpoints to the setpoint-controller at the appropriate time of the day.

Database: This is a stand-alone application that subscribes to all EWH applications registered at the NSCL and stores their inbound data in a SQL database. This database is independent from the built-in database provided by the NSCL where the REST resources are persisted. It is therefore a non-critical component. The motivation for its implementation is two fold: (1) to provide data redundancy seeing as though the OM2M development was still under Beta release, and (2) to provide fast data access for analysis purposes.

The NIP, web application, scheduler, setpoint-controller and database application were designed and developed by the author.

3.1.3 RESTful architecture and runtime operation

Having defined the role of each component in System A, the operation of each is now described. Specific focus is placed on the RESTful operation and architecture of the system as this is at the core of the SmartM2M standards and therefore the contribution of this work. Refer to the SmartM2M specification overview presented in Section 2.2.1.2 (page 15).

Figure 3.2 depicts a timing diagram for a typical full transaction from EWH to Scheduler with the NSCL as central point of interface. The corresponding RESTful resource tree residing on the NSCL during runtime is depicted in Figure 3.3. The reader is reminded that this resource tree is an application specific *implementation* of the standard RESTful architecture proposed by SmartM2M, which was introduced in Figure 2.4. Referring to each of these two figures, a discussion on the components of System A follows below.

EWH client:

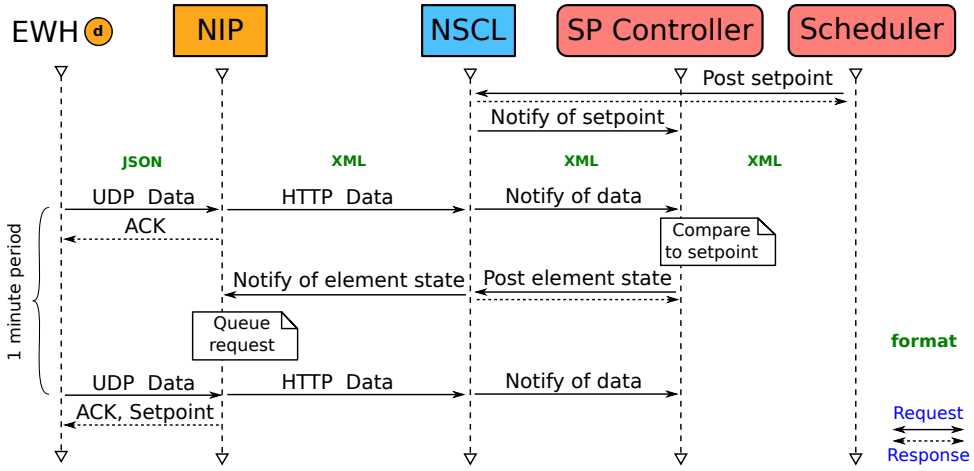


Figure 3.2: Timing diagram of a typical end-to-end transaction involving all the components of System A.

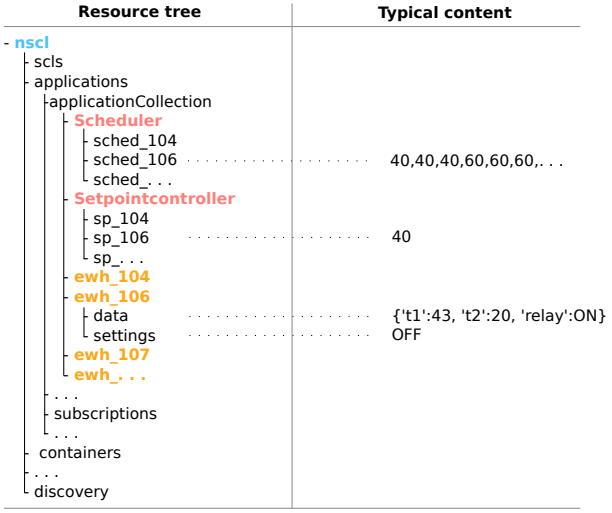


Figure 3.3: System A application specific implementation of the RESTful resource tree architecture proposed by SmartM2M. The resources reside on the NSCL, and shows *ewh*, *scheduler* and *setpoint controller application* along with typical document content at the right (cross-reference with Figure 2.4 on page 17).

Runtime operation (Fig 3.2): The modem operates in *transparent character pipe* mode, which means that it is pre-configured with the IP address and port of the NIP and, on start-up, automatically sets up a duplex character pipe session via the serial line. Due to the constraints of *transparent character pipe* mode, a simple custom protocol between the EWH client and the NIP was designed with the following characteristics:

- Once every minute, each Mk2 EWH client posts a JSON formatted data-stamp to the NIP e.g. `{"t1":43, "t2":30, "relay":"ON" ...}`.
- Upon reception of a data-stamp, the NIP responds with an acknowledge (also using JSON), which is used by the device to confirm connectivity e.g. `{"status":"ACK"}`
- The UDP network protocol is used, meaning that the interaction between the EWH and NSCL is *connectionless*.

- In order for the server to send a command back to the EWH device, it is queued at the NIP, and *piggy-backed* with the acknowledge message.

Each data-point is 219 bytes long, which sums to roughly 10 megabytes per EWH per month.

RESTful architecture (Fig 3.3): Each EWH has an `application` resource registered at the NSCL, which is created on its behalf by the NIP (shown in orange). Under the `application` resource there are two `container` resources; (1) a `data` container where the inbound measurement stamps are cached for use by the other applications, and (2) a `settings` container to which the NIP self-subscribes in order to receive command outbound for a EWH device. If a notification is received from the `settings` container, the command is queued at the NIP, and *piggy-backed* the next minute an EWH delivers a data-stamp.

NIP:

Runtime operation (Fig 3.2): The NIP listens on a single UDP port for any inbound data-stamps from EWH clients. Upon reception of a data-stamp, the NIP parses the EWH ID from the stamp, and POSTs the data-stamp to the `data` container of the corresponding EWH `application` at the NSCL. When an *unknown* EWH reports in, a new EWH `application` resource is created. The NIP also delivers any cached commands for the EWH, *piggy-backed* on the ACK message. As mentioned, the NIP receives commands for the EWH by subscribing to the `settings` container of each EWH `application`.

RESTful architecture (Fig 3.3): In line with the SmartM2M philosophy, the NIP acts as a *transparent* agent between the EWH devices and NSCL. It it therefore not registered as an application at the NSCL. Its purpose is only to enable the NSCL to *perceive* the EWH devices as fully SmartM2M compliant. All individual messages between the NIP and Mk2 clients are designed to be transported within a single UDP IP packet which avoids the need for packet reconstruction and greatly reduces complexity. This design means the NIP can be single threaded and process each request synchronously, making it easy to implement but not scalable. Even though scalability is not a requirement for System A, this design of the NIP can however be horizontally scaled by simply deploying them in *parallel* and dividing the Mk2 units equally among the proxies.

Setpoint controller:

Runtime operation (Fig 3.2): The Setpoint controller subscribes to the `data` container of each EWH and thereby receives a push notification from the NSCL on each new data-stamp. It then compares the data-point to a configured setpoint, and if the current temperature of the EWH does not match the desired temperature, it POSTs a new relay state to the `settings` container of that EWH.

RESTful architecture (Fig 3.3): The Setpoint controller registers a single `application` for itself at the NSCL, and creates a container for each EWH under that `application` resource in the format `sp_<ewhID>`. Each container simply holds a single temperature value which is the reference temperature value to which an inbound data-stamp is compared to.

Scheduler:

Runtime operation (Fig 3.2): The scheduler application wakes up every 15 minutes and POSTs a new set-point value from a list to each `settings` container of the setpoint controller application.

RESTful architecture (Fig 3.3): Similar to the setpoint controller the scheduler also creates a container for each EWH in the format `sched_<ewhID>`. This container holds a *list* of 96 temperature values (one for every 15 minutes in 24 hours) which represents the configured temperature schedule of each EWH.

3.1.3.1 Interface applications

The database-, web- and mobile application are not registered as `applications` on the NSCL, since they do not provide any data of their own. They only interact with the data hosted by the above applications. These applications represent the typical 3rd party "data sink" application that can be readily integrated into the system, owing to the standardised SmartM2M API.

Database: The database subscribes to the `data` container on each EWH *application* resource and thereby receives notifications from the NSCL and stores each inbound datapoint in an SQL database.

Web application: The web application is used to display the data inbound from the EWH, as well as to set the temperature schedule for each EWH device. Setting the schedule on the web interface, translates into a NSCL POST request, which updates the corresponding `sched` container of the scheduler `application`. The web application is only intended to be an administrative interface and not a general user portal.

3.1.4 Implementation: IT infrastructure

The NSCL, NIP Scheduler, Setpoint Controller, Database and Webserver are all hosted on the same virtual machine, running CentOS 7 with 2 Gb of memory, 42 Gb of storage an a Intel(R) Xeon(R) CPU E5-2690 processor at 2.90GHz. Two ports were made publicly accessible, one for the NIP receiving inbound UDP packets from the EWH clients, and one for the web application.

Developer	Responsibility	Contact
Nico Naude	Mk3 firmware (Sensor and actuator drivers)	nnaude@gmail.com
Andrew Cloete (author)	Mk3 firmware (Modem driver using opaque mode)	andrewhcloete@gmail.com
Lourens Visagie	Mk3 firmware (Bootloader for OTA)	lourensv@gmail.com
Robert Sandell	DTL, Web server, Admin server, Database engineer, Load testing client emulator	rcsandell@gmail.com

Table 3.1: System B work acknowledgements

3.2 System B: Lightweight protocols and scalability

3.2.1 Requirements and constraints

System B was developed as a pilot project consisting of 450 units. The scale of the contract meant that the requirements of System B were significantly more rigorous than that of System A. Specifically, the emphasis was placed on using lightweight protocols for low data costs as well as end-to-end scalability and robustness. Additionally, for increased reliability, flexibility and security, *over-the-air* (OTA) firmware upgrades to the Mk3 controller was also made a requirement. To satisfy the above requirements, the *transparent character pipe* AT command mode of System A could no longer be used. Instead *opaque* AT command mode was used, the application of which is discussed in detail in this section.

The OM2M platform was discarded in System B for its lack of maturity. Instead it was decided to use tried-and-tested *web* technologies for their robustness, security and scalability. Furthermore M2M application layer *protocols* are more mature than the more recent M2M *middleware frameworks*. Specifically, the MQTT protocol was used for its light weight, simplicity and rich ecosystem of software implementations. It was therefore decided to shift the focus from M2M frameworks, down one layer, to M2M application protocols.

3.2.2 System B layout

Figure 3.4 shows the layout of System B. The author would like to clearly denote his own work, and acknowledge the developers that contributed in building System B. Referring to Figure 3.4, the scope of the author's work lies only at the *EWH client* and *MQTT broker*. Specifically, for System B, the author was only responsible for (1) selecting the MQTT protocol (and Mosquitto broker) as Internet interface for the EWH clients and, (2) developing the communication software interface of the modem on the Mk3 hardware. This is in line with the project objectives, since the goal is to determine the appropriateness of available M2M application layer protocols. Consequently, this section will only focus on these two points. A brief overview of the complete system follows in the next section, to provide an understanding of its function, and to point out the differences with System A. Table 3.1 gives acknowledgement to all persons involved, denoting the part of the system they were responsible for. As before, a discussion off each component follows below.

EWH client: The hardware constitutes the **Mk3** EWH controllers developed by Brown [25] as discussed in Section 2.4 (page 28). This hardware implements the same functionality as Mk2, but the cellular modem has been replaced with a Ublox LEON-G100, and the microcontroller with the an Atmel Xmega128A4U. Contrary to Mk2, this microcontroller was programmed using standard C, which allowed software flexibility and a wide range

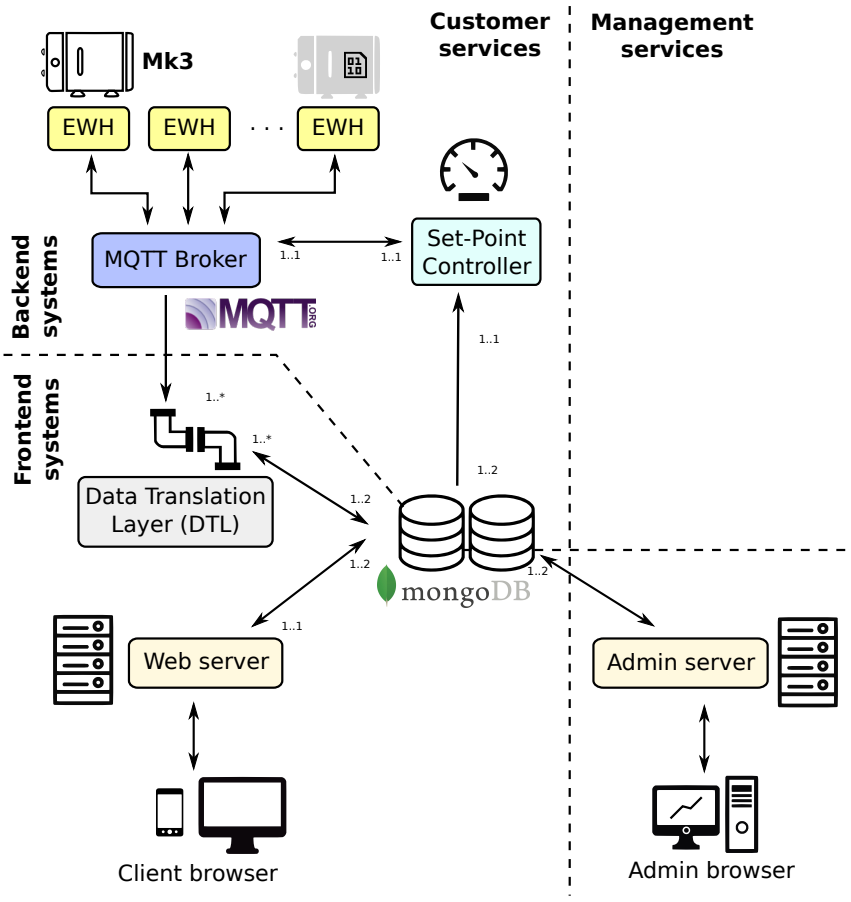


Figure 3.4: Architecture of System B showing MQTT interface with popular web technologies

of external libraries. For this reason the *opaque AT command* mode was adopted for modem interface, which allows the use of binary data making application protocols such a MQTT viable. This mode is however considerably more complicated from a software perspective. The communication software developed for the Mk3 hardware is discussed in depth in the next section. As in the case of System A, a simulator was implemented to test the end-to-end system with mock-up data.

MQTT broker: This is the Mosquitto open-source implementation of the MQTT broker specification [109]. Where the NIP and NSCL formed the core of the M2M functionality in System A, the MQTT broker forms the core of System B. Unlike the OM2M platform, the Mosquitto broker is part of a rich ecosystem of libraries and contributors.

Data-translation-layer (DTL): The DTL is a custom-built application that proxies the interface between the MQTT broker and the database. The DTL is a listening client on the MQTT broker subscribed to *all* EWH activity. The main purpose of the DTL is to capture the data-stamps from all EWHs and to store them in the database.

Setpoint controller: Similar to System A, the purpose of the SP controller is to monitor all EWH temperatures and dispatch commands in the event that temperatures are not satisfactory. In System A, the reference set temperature was stored as a REST resource on the NSCL. In system B, the setpoint application interfaces directly with the

database to obtain the reference set-point for each EWH. The SP controller also performs the scheduling functions.

Web applications: The front-end web applications provide client and administration orientated system access capabilities. The web applications are served through NodeJS [110] and written using the Meteor [111] single page application (SPA) client side frameworks optimised for mobile devices. Screen captures of the client and administrator web interface may be found in Appendix C.

MQTT topic path configuration: The principles of MQTT topic paths were discussed in Section 2.2.3.1 (page 19). MQTT topics designed for System B are as follows: Each Mk3 controller *publishes* a periodic time stamp under the topic "`ewh/ts/<ID>`", where the "`<ID>`" is the IMEI number of the modem, which is unique to each device. Each Mk3 controller also *subscribes* to the topic "`ewh/sp/<ID>`" so that it may receive asynchronous notifications via the broker. The SP controller *subscribes* to all timestamps under the topic "`ewh/ts/#`", and can publish set points to EWH using the "`ewh/sp/<ID>`" topic. The DTL *subscribes* to the root topic `ewh/#` using a wildcard and therefore receives notifications on *all* EWH related messages. The DTL uses topic sub-path to interpret the message content. This design demonstrates that the use of topics is lightweight yet provides a powerful mechanism for establishing one-to-one, one-to-many, and many-to-many relationships.

3.3 Communication software stack design: Mk3

This section presents the design and implementation of the communication software stack for the Mk3 hardware. The purpose of the communication stack is to abstract and hide the functions of the modem and present the feature application with a high-level API, which enables the sending and receiving of messages.

3.3.1 Stack overview

Figure 3.5 gives an overview of the layers of the communication software. The software stack consist of five layers each with an API used by the layer above it. Below follows a discussion of each layer from the bottom up.

L1 - Serial layer: This layer abstracts the UART module of the Atmel processor. It is responsible of buffering bytes incoming from the modem and transmitting bytes to the modem.

L2 - AT layer: This layer is responsible for reading the inbound buffer of L1, parsing AT responses from the modem. As part of the parsing process the AT responses also need to be classified (discussed later). This layer is also responsible for serialising data into the appropriate AT command character strings, which can then be passed to L1 for transmitting to the modem over the serial line.

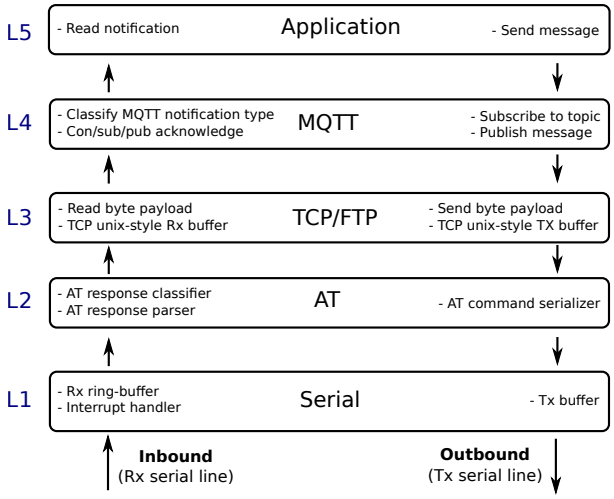


Figure 3.5: Overview of modem driver software stack for Mk3 hardware showing 5 layers of software abstraction.

L3 - TCP: This layer provides a TCP API similar to that of the standard UNIX C socket interface, but instead of using the IP stack present in most operating systems (which is not available in this case), it employs the TCP specific AT commands from L2. In other words, it is essentially a Unix-lookalike wrapper for TCP communication that uses underlying AT commands to perform the connection.

L4 - MQTT: This is the open-source, 3rd party MQTT embedded C library by Paho [87]. It interfaces with L3, to send and receive MQTT messages. The role of this layer is to interpret the binary MQTT headers from the inbound messages to determine the message type, as well as to add MQTT headers to outbound messages for correct interpretation at the broker.

L5 - Application This is the upper most layer of the feature application where the business logic of the device is executed. At this layer the complexities of the communication software stack are hidden and presented simply as a *send-message*/*get-message* type API.

3.3.2 Software design

Figure 3.6 presents a high-level overview of the communication stack software structure. The software modules are represented as clear boxes and the grey boxes give some of the main features of each module. The layout of the modules roughly coincide with the communication stack in Figure 3.5. A discussion on each component follows below.

3.3.2.1 Serial driver

Inbound bytes from the Rx UART line are *read* using an interrupt handler routine that simply pushes each byte to a ring buffer of size 256 B. The driver also provides an API function to write bytes to the modem. The modem UART operates at 9600 baud.

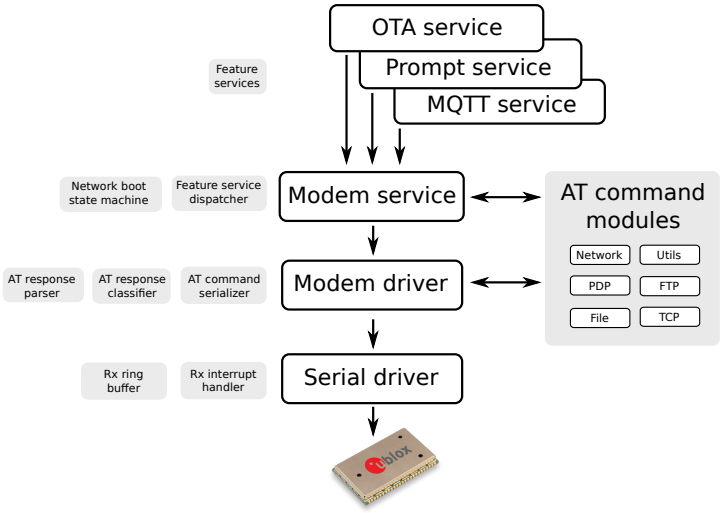


Figure 3.6: Overview of modem driver software modules for Mk3 hardware

3.3.2.2 AT command modules

The AT command modules form the core of the communication stack. In essence, these modules present a software wrapper for AT commands to provide a uniform and systematic way of performing three key modem operations:

1. Serialise AT commands, populating the command string with the appropriate variables.
2. De-serialise AT responses, parsing the appropriate string values as software variables.
3. Provide an API to *get* and *set* the above variables appropriately.

Each module groups AT commands together that perform similar functions. A discussion on each module along with some of the key AT commands it implements follows below. A complete reference of all the AT commands used can be found in Appendix D (page 97).

Network: AT commands that perform basic cellular network configuration and interaction, such as reading the current network operator name, requesting signal strength or dialling a USSD code to get the current SIM balance.

PDP (packed-data-protocol): In opaque AT command mode, the modem performs a dial-up *internally* and establishes a PDP session with the tower. The AT commands in this module configures and controls the PDP context. This includes setting the Access Point Name (APN) and starting, and stopping the PDP session. The internals of the ongoing PDP session is completely hidden from the controller in opaque AT command mode.

Utilities: Basic commands that perform internal commands, such as reading the device clock and IMEI number.

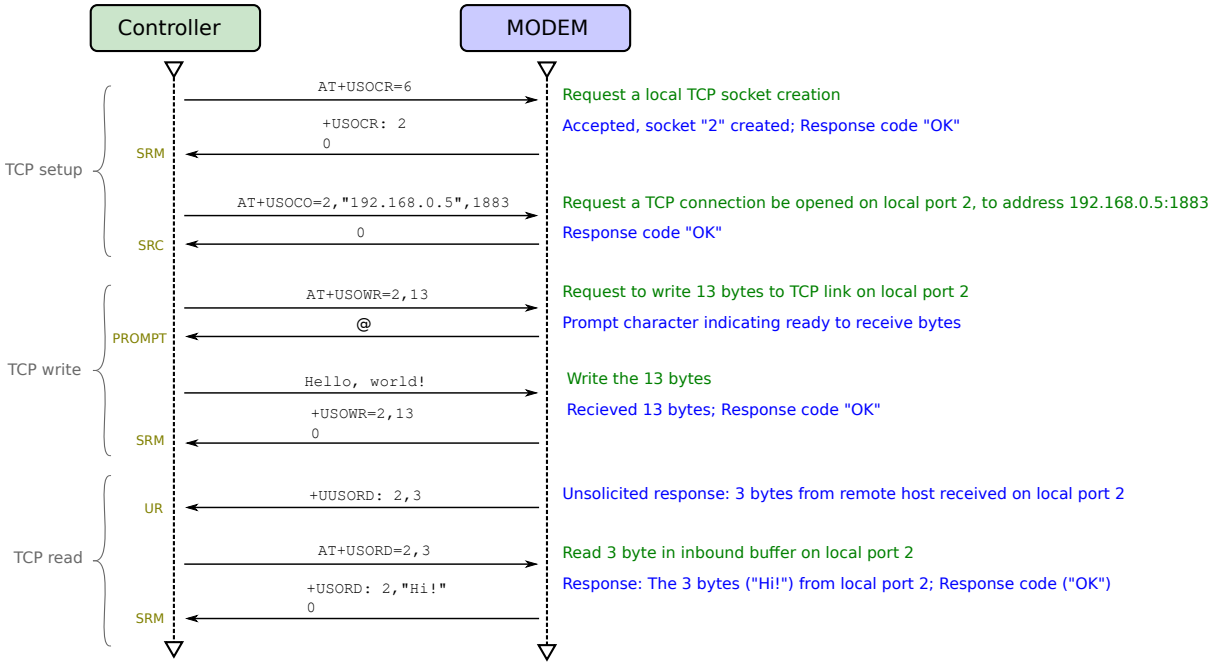


Figure 3.7: AT-TCP command interface showing an example *"Hello, world"* transaction.

TCP: These AT commands form the basis of the TCP layer described in the previous section (3.3.1). Figure 3.7 presents a timing diagram of an example *"Hello, world"* AT-TCP transaction, showing the *setup*, *writing* and *reading* procedures. The text on the arrows show the actual characters being transmitted and received. The text on the right provides a description with corresponding colours. The text in yellow indicate the *type* of AT response received (all four types are shown in this example and discussed later).

FTP: The LEON G100 modem also supports the FTP protocol. The FTP AT commands are used by the OTA service to download new firmware byte-code, which is stored in the 1 Mb internal flash memory of the modem. This is then later used by the bootloader to reprogram the device.

File: For manipulation of the files on the modem. Specifically to delete an outdated firmware file.

3.3.2.3 Modem driver:

The purpose of the modem driver module it to ensure that all transactions with the modem occurs appropriately within the boundaries of the modem's specification. This includes performing the following basic tasks:

- Determine the current modem state (cellular and Internet connectivity status).
- Ensure appropriate waiting periods between AT commands are enforced.
- Ensure unsolicited responses are caught.
- Ensure all AT responses are parsed and classified appropriately.
- Detect error conditions and respond accordingly.

- Provide an API to the upper layers for executing AT commands and returning AT response results in a uniform and systematic format.

AT response parser: A noteworthy part of the modem driver is the mechanism for parsing and classifying inbound AT responses. In opaque AT command mode, there are four different types of AT responses (1) a *solicited response code (SRC)*, (2) a *solicited response message (SRM)*, (3) an *unsolicited response (UR)* and (4) a *prompt*. An example of each was shown in Figure 3.7. As the name suggest, solicited responses are the synchronous result of prompting the modem, whereas unsolicited responses are messages pushed by the modem asynchronously when an important external event occurs (e.g. receiving of an SMS). The response types are as follows:

- Solicited Response Code (SRC): consists of a single character code to indicate success (`"0"`) or failure (`"4"`).
- Solicited Response Message (SRM): consists of a multi-lined message containing a response payload, followed by an SRC to indicate success or failure.
- Unsolicited Response (UR): may or may not be a multi-lined message. Has no response code since it is not apropos of an AT command.
- Prompt: An `"@"` character indicating a prompt for byte data.

This is mentioned here to highlight that the AT interface is inconsistent and contains many exceptions. In fact, the Ublox application notes suggests treating each AT response as unique [112, pg.10]. The parsing and classification of AT responses therefore requires a *decision making tree*, which employs a combination of timeouts and character matching conditions. Furthermore, this complexity is *vendor specific* and cannot necessarily be used for different modems. The decision making trees of the AT response parser and the modem "next state" evaluator may be found in Appendix E.

3.3.2.4 Modem service

The modem service module forms the point-of-contact of the modem functionality with the rest of the feature application. It is responsible for (1) performing a start-up sequence that ensures the modem is connected and ready, and (2) dispatch and schedule higher level services that use the communication stack (in this case the MQTT-, Prompt- and OTA service).

3.3.2.5 MQTT service

The MQTT service is loaded by the modem service and it provides an API to the feature application for sending messages and receiving notifications. The service is also responsible for setting up and managing the connection with the MQTT broker, ensuring that all handshaking with the broker proceeds correctly. The MQTT service decision making tree may be found in Figure E.3 in Appendix E.

The Paho MQTT 3rd party embedded C library [87] is used to build and interpret MQTT packets. QoS level 1 is established with the broker, meaning that any publishing of messages receives a publish-acknowledge (PUBACK) from the broker. The MQTT service also *subscribes* to the MQTT broker under its own ID in order to receive notifications for use by the prompt service.

		A	B
Methodology	M2M focus	Middleware standards	Lightweight protocols
	Scale	10 clients	450 clients
Design	Architecture	REST	Custom
	Application layer protocol	Custom	MQTT
	Transport layer protocol	UDP	TCP
	Data presentation	JSON	JSON
	Data semantics	Custom	Custom
Controller	Interface	NIP	MQTT broker
	CPU	Particle Core	Atmel Xmega 128A4U
	Modem	Sierra Wireless Maestro 100 EVO	U-blox LEON G100
	AT mode	Transparent character pipe	Opaque
	Communication stack	Simple	Complex
	Request mechanism	Synchronous	Asynchronous
Controller request features	Set valve state	✓	✓
	Set relay state	✓	✗
	Set temperature setpoint	✗	✓
	Get airtime balance	✗	✓
	Get network provider	✗	✓
	Get signal strength	✗	✓
	Get cell number	✗	✓
	OTA update	✗	✓

Table 3.2: Summary of differences and similarities between System A and B

3.3.2.6 Prompt service

The purpose of the prompt service is to allow a remote MQTT client to perform requests, like the current airtime balance, cellphone number and RSSI quality. The prompt service listens for inbound MQTT notifications, interprets the command, dispatches the appropriate action and sends the results back to the broker.

3.3.2.7 OTA service

The over-the-air (OTA) updated service uses the FTP AT commands made available by the modem service to set up an FTP connection to a remote server and download new firmware binary code. The OTA service gets dispatched by the prompt service when an OTA notification is received along with an address of the file server. FTP login credentials are hard-coded to the device. After the firmware files have been successfully downloaded, control is handed over to the bootloader (see Table 3.1), which performs an integrity check and writes the new software to flash memory.

3.4 Closing remarks on System A and B design

Table 3.2 summarises the main differences and similarities between System A and System B. In System A, the prominent component is the NIP, which needed to be developed from the ground up for this study in order to enable standard compliant interface of the Mk2 controller with the NSCL. In System B, the prominent component is the communication software stack for the Mk3 controller, which was also developed from the ground up for this study in order to enable interface with the MQTT broker. In the next chapter, the operation of both systems is evaluated.

Chapter 4

End-to-end M2M system results

This chapter discusses the findings of implementing System A and B towards evaluating Objective 1. The goal is to determine the strengths and weaknesses of each system in order to ascertain its applicability as an IoT solution for EWH monitoring and control. Comparing System A with B to determine a "*trump*", is not an informative exercise, since they were designed with diverging requirements and applications in mind. Instead, comparison is made primarily to highlight and contrast the M2M strengths and weaknesses in each.

4.1 System A efficacy

System A was deployed as a pilot study and consisted of six *field* Mk2 controllers installed in households the Cape Winelands district, and one *lab* Mk2 unit installed on a laboratory EWH installation for testing purposes. The system was in operation for ten months during which it collected approximately 3.8 million data points.

Figure 4.1 shows a plot of EWH temperatures for one household over 24 hours with element state for reference (top), as well as a balloon plot of the hot water usage events during that same day (bottom). Detail concerning the events is presented in Table 4.1. The energy cost per event was estimated by calculating the *effective* energy of the consumed water using the volume and temperature of each event (at R1.50/kWh).

Event time	Volume (litres)	Duration (minutes)	Temp (°C)	Energy (kWh)
07:18	42.0	5	48.8	1.23
08:00	61.4	7	51.4	1.96
19:37	37.9	4	49.7	1.05
19:43	43.4	3	49.3	1.23
21:10	95.6	6	50.3	2.93

Table 4.1: Summary of event breakdown of 24 hours of data collected for one EWH using System A

Figure 4.2 shows the result of an experiment conducted on the laboratory EWH (using System A) over 24 hours to validate the correct operation of the Scheduler and Setpoint controller. The Setpoint controller controls against the outlet temperature. To

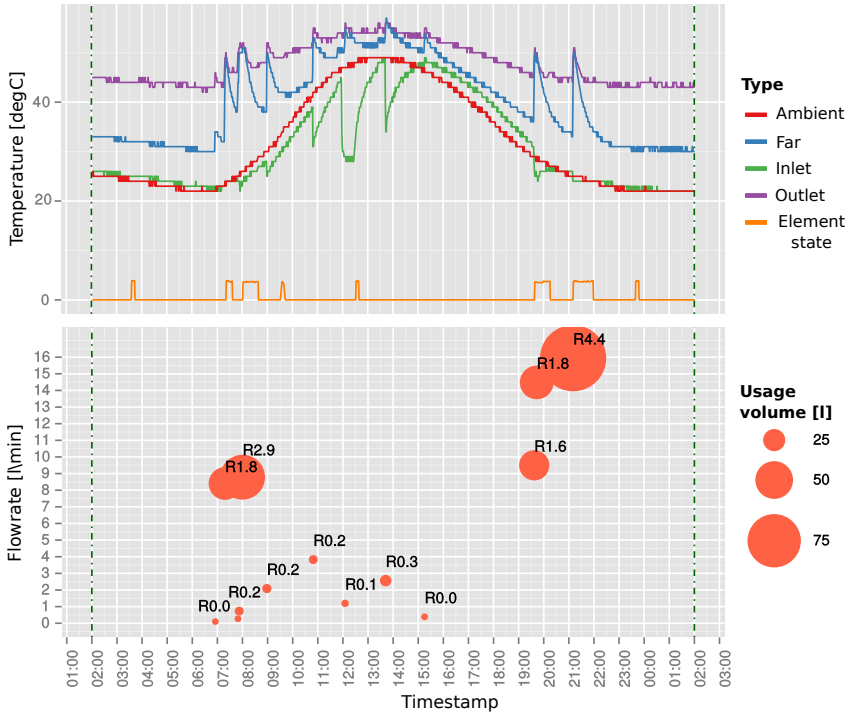


Figure 4.1: Temperature and event data collected for one EWH on 28 October 2015 using System A.

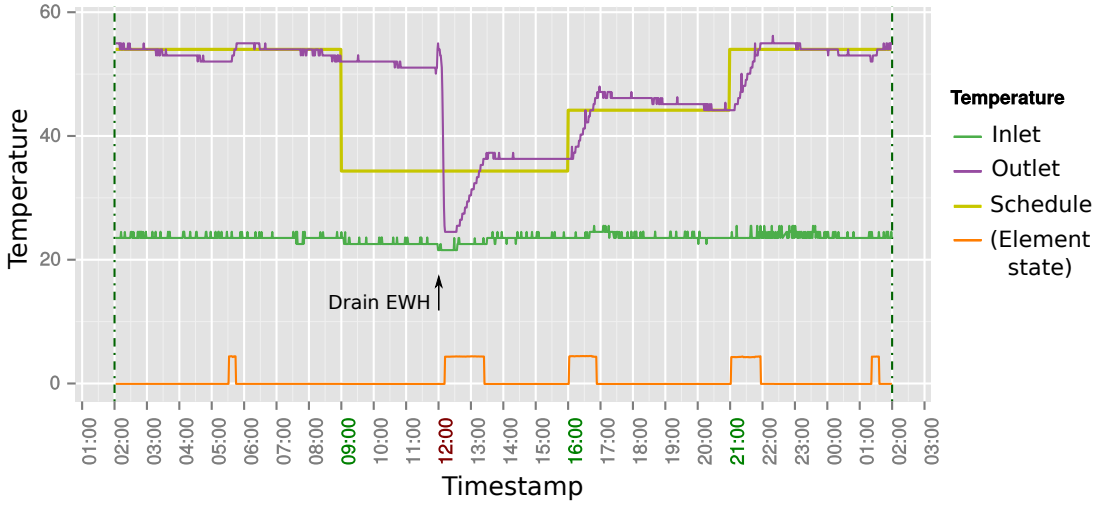


Figure 4.2: Laboratory experiment showing schedule control operation over 24 hours

emphasise the control, the EWH was drained for 25 minutes at 12:00 to cool the EWH down. After the event, the figure clearly depicts the outlet temperature following the scheduled temperatures. Note that Setpoint controller has hysteresis of 2°C around the reference point.

4.1.1 Collected data integrity

Since a data-point is reported every minute, a total of 60 points are expected from each EWH hourly. This number can be used to calculate a *packet delivery rate (PDR)* performance indicator as a percentage of the total received data points. Figure 4.3 gives

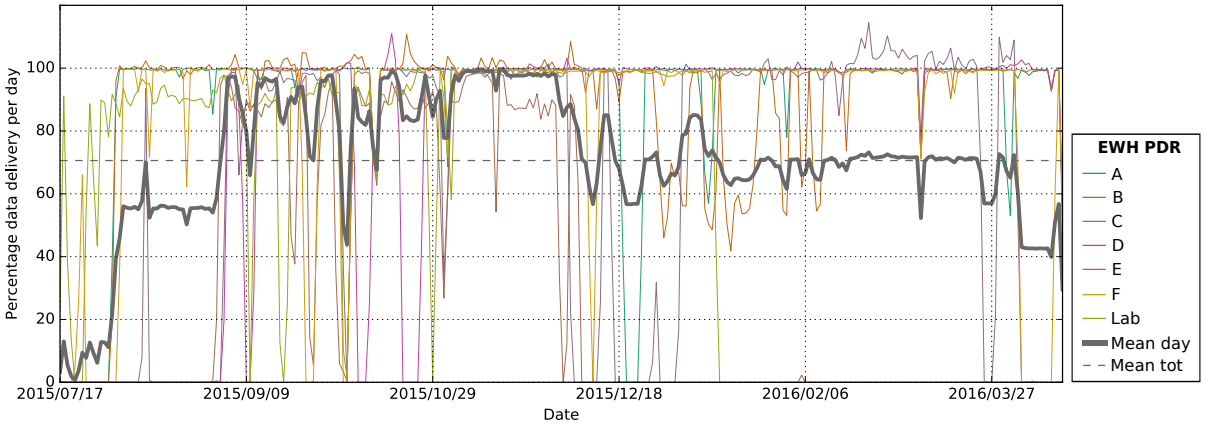


Figure 4.3: Packet delivery performance of System A over 10 months. Individual and average delivery performance.

a plot of the daily delivery performance of the seven Mk2 units over a ten month period. The average PDR shown is 70.6% over the period. From this it is shown that collecting continuous data over a relatively long period proved to be a challenging task. This is due to three reasons: (1) lack of ability to acquire SIM airtime balance and therefore unable load in advance, (2) participants switching off the power supply to their EWHs, and (3) the modem truncates data-point messages.

On several occasions the airtime depleted unexpectedly causing the Mk2 units to stop reporting data. This resulted in roughly 12 hours of down-time until the problem was identified and new airtime loaded. It was also found that some participants switched off their EWH to save electricity. In this case the participants were contacted and asked to keep their EWH on.

Intermittent data loss was also caused by the fact that the modem was used in *transparent UDP* mode. In this mode it is impossible to specify to the modem how the outbound payload should be serialized as a UDP packet. In most cases the data string of 219 bytes was packed correctly as a single UDP packet, and arrived at the NIP intact. However, the modem would occasionally split the data string and send it as *two* UDP packets, which then is interpreted by the NIP as 2 malformed JSON strings. This corruption of the data-point occurred on average once per hour per EWH unit. This implies that only around 2% of data loss could be attributed to this error. The majority of EWH down-time was due to the logistical challenges listed above.

4.1.2 System A efficacy conclusion

From Figures 4.1 and 4.2 it is clear that System A proved effective and able for the task it was designed for. The system and data was sufficient to manually select periods (10 days) of complete datasets for each EWH to conduct a preliminary study on the savings possible due to schedule control. This study is presented in [113].

The logistics of deploying and managing the Mk2 units proved to be the largest stumbling block for System A. Being able to monitor airtime balance feedback is a crucial requirement that was overlooked. Providing users with a basic interface is also important since this motivates them not to bypass the system. Finally, even though UDP

data truncation only resulted in a 2% data loss, it still implied inherent unreliability in communication, which is a point of concern for any data critical applications.

4.2 System B efficacy

System B was deployed and at the time of this writing, consisted of 60 Mk3 deployed to EWHs in the Mkhondo district municipality region, and 30 in the Cape Winelands region. Following the form of Chapter 3, the focus of the results will mainly be concerned with the communication stack of the Mk3 hardware and the appropriateness of the MQTT protocol and the Mosquitto broker.

The end-to-end system functioned as expected and from an operational perspective displayed a significant improvement compared to System A. Specifically, by using the modem in *opaque* AT command mode and employing the MQTT protocol, enabled more functionality and improved EWH client connection reliability and performance. Appendix C gives selected screen captures of the client and administrator web interface.

4.2.1 Communication driver for Mk3

The implementation of the communication driver took roughly three months to implement and refine, making it the largest single application presented in this research report. Table 4.2 gives the serial output of the Mk3 unit for three minutes from start up. The time in seconds and milliseconds is given in column 2. The rounded *delta* time is given in column 3 to indicate duration longer than one second. Column 4 gives the serial message string output via the debug serial port of the Mk3 device. A discussion on the rows of interest follows below:

1-2: Communication driver starting by performing a hard reset on the modem by pulling the reset line low.

3-4: After hard reset, modem is given 40 seconds to establish connection with the cellular access network and perform all layer two handshaking.

5-6: Send `"AT"` to ensure serial communication is operational.

7-18: Send sequentially several AT commands to read system and network details such as IMEI number, network provider, SIM data balance *et cetera*.

19-22: Setup Packet-service-data and start an internal Packet-data-protocol (PDP) session. Notice that the PDP session takes around 30 seconds to establish.

23-24: Resolve the host IP address, using the resolve AT command. In this case the IP address is used but a domain name may also be used.

25-28: Setup and establish a TCP connection to the resolved IP address.

29-31: Send a *connect* request to the MQTT broker (not shown), receive a 4 byte reply in the form of a URC, prompt a read and interpret as a *connection acknowledge (CONACK)* response.

	Time	ΔT (sec)	Output
1	0:00.147		Running. Ver: 1.5.4
2	0:00.155		@@@@@@@@@@ HARD RESET on state: {IDLE} @@@@@@@@@@
3	0:02.670	3	Modem warming...
4	0:43.257	41	Modem warm
5	0:43.261		Checking AT...
6	0:43.292		AT OK
7	0:43.295		Read IMEI...
8	0:43.480		IMEI 352432065551607
9	0:43.483		Getting RSSI...
10	0:43.630		RSSI 28,99
11	0:43.634		Reading COPS...
12	0:43.799		Operator VodaCom-SA
13	0:43.802		Clock...
14	0:43.969		Clock 16/09/29,16:02:23+08
15	0:43.973		Getting number...
16	0:46.574	3	Number 27609063512
17	0:46.578		Getting balance...
18	0:50.894	4	Balance 38.75 MB
19	0:50.897		Set PSD...
20	0:50.936		Set PSD OK
21	0:50.939		Act PDP...
22	1:22.589	32	Act PDP OK
23	1:22.594		Resolve... 52.31.251.102
24	1:22.789		Resolved 52.31.251.102
25	1:22.792		TCP setup...
26	1:22.940		Local port 0
27	1:22.945		TCP connect... 52.31.251.102 : 1883
28	1:23.606	1	TCP CONNECTED!
29	1:24.659	1	URC: {+UUSORD:}
30	1:24.664		TCP read: {AT+USORD=0,4
31	1:24.823		}CONNACK
32	1:25.578	1	URC: {+UUSORD:}
33	1:25.582		TCP read: {AT+USORD=0,5
34	1:25.740		}SUBACK
35	2:00.280	35	{"ver":"1.5.4","SP":60,"T1":10,"T2":9,"T3":11,"T4":10,"R":"On","time":119}
36	2:01.442	1	URC: {+UUSORD:}
37	2:01.447		TCP read: {AT+USORD=0,4
38	2:01.607		}PUBACK
39	2:02.361	1	URC: {+UUSORD:}
40	2:02.365		TCP read: {AT+USORD=0,34
41	2:02.556		}MQTT notif: {"sp":0}
42	2:02.559		SP: 0
43	3:01.238	59	{"ver":"1.5.4","SP":0,"T1":10,"T2":10,"T3":11,"T4":10,"time":179}
44	3:02.318	1	URC: {+UUSORD:}
45	3:02.323		TCP read: {AT+USORD=0,4
46	3:02.484		}PUBACK

Table 4.2: Mk3 serial output from start-up up until second reported data-point

32-34: Send a *subscribe* request to the MQTT broker (not shown), receive a 5 byte reply in the form of a URC, prompt a read and interpret as a *subscription acknowledge (SUBACK)* response.

35-38: Publish first data-point. Receive a 4 byte reply in the form of a URC, prompt a read and interpret as a *publish-acknowledge (PUBACK)* response

39-42: Receive a 34 byte URC and prompt a TCP read. A JSON string has been received and is passed the the Prompt Service, which sets the local setpoint to $0°C$.

43-46: Publish second data point. Take note that the "SP" is now updated. Again, receive a *publish-acknowledge (PUBACK)* response.

Using the communication stack driver described in Section 3.3, the Mk3 microcontroller was able to perform all the necessary actions on the LEON G100 modem to connect to the MQTT broker via the cellular access network.

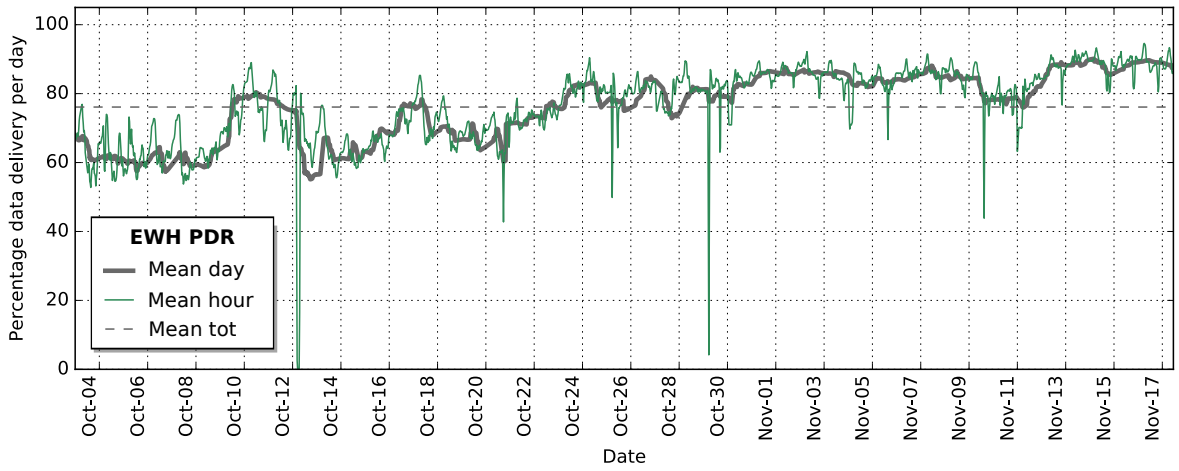


Figure 4.4: Overall network packet delivery rate consisting of 71 Mk3 units for October and November 2016.

4.2.2 Collected data integrity

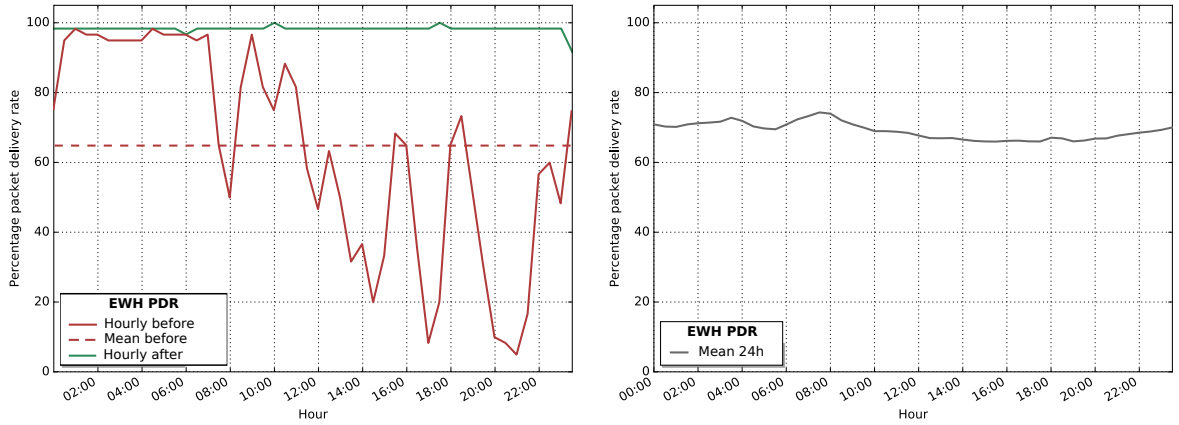
Similar to System A, the Mk3 units report once every minute, therefore the expected packet delivery rate (PDR) for each EWH is known. Figure 4.4 shows the average PDR performance of all the EWHs over one month. The figure shows both the hourly and daily PDR, and indicates that on average, 78% of the expected data is received. It was found that units installed in the Cape Winelands performed excellently, suffering little to no data-loss. It was therefore surprising to find that Mk3 units installed in Mkhondo suffered severely from data loss, decreasing the average network performance (as was shown in the figure). Contacting the mobile provider revealed that the 2G service in Mkhondo had been allocated limited bandwidth. After switching to a different mobile provider, data delivery performance increased significantly. Figure 4.5a shows the data delivery performance of one Mk3 installation in Mkhondo before and after switching network providers. Before the network switch was made the device lost connectivity on average within 15 minutes of connecting to the MQTT broker, and took on average 10 minutes to reconnect. The RSSI index reported from this unit in Mkhondo was 80% (between -69 and -57 dBm gain), confirming that the data loss was not due to low signal strength but rather poor quality-of-service from the network provider.

It was interesting to note that the network performance of the Mkhondo units changed during the day, doing better in the early morning hours, peaking at 6 AM, and poorly in the afternoon (see Figures 4.4 and 4.5a). Figure 4.5b gives the **mean** hourly PDR during a day for the month of October, showing slightly better PDR on average during the morning hours.

Finally, it was again found that some EWH client participants switched the power supply to their EWH off for certain periods of the day, even though they had full access to the client web application, which enabled them to set a heating schedule.

4.2.3 System B efficacy conclusion

From the above, is it concluded that System B proved effective and it met the design specifications. Although System A and B should not be directly compared, System B proved more effective from an operational perspective than System A. This is due to thee



(a) Single EWH before and after switching provider

(b) Mean PDR over 24 hours for 23 days in October 2016

Figure 4.5: Hourly packet delivery rate performance (PDR) over 24 hours

factors: (1) using software with large supporting communities (i.e. MQTT broker and client) relieves development burdens, (2) using opaque AT command mode is more robust and enables more features (lightweight protocols, OTA upgrades), and (3) implementing "lessons-learnt" from System A such airtime balance requests and client interface.

4.3 Discussion on M2M implementation

4.3.1 Evaluation of the SmartM2M standardised interface

The standardised interface provided by the SmartM2M standards (System A) provided the capability for rapid incorporation of external applications. The *set-point controller* and *scheduler* are examples of applications that require no internal knowledge of the EWH units to perform their function. Furthermore, the *scheduler* can be replaced (for example) with *predictive controller* that learns usage behaviour by subscribing to the individual EWH applications. This demonstrates the advantage of a standardised interface. In contrast, the DTL of System B represents the type of application that the oneM2M standards aims to prevent since it is a typical *vertical* application, with a custom interface. Hence, it must be stated that although the RESTful resource tree and its interaction has been standardised, the internal data format of the instance documents has not been. In other words, at this point in time SmartM2M is only proposing a standardised framework and not a unified data model for inter-domain *semantics*. In the case of System A, a simple JSON document is used to carry the data-point, which requires that external applications need to be *preprogrammed* to interpret that data fields.

The adolescence of the M2M standards means that it has yet to see large scale adoption. This has the disadvantage that community support and example implementations are limited. Comprehension of the standards via the primary specification documents was therefore a daunting task. Nevertheless, from System A, the standards prove promising and are strongly advocated for future SG applications.

4.3.2 OM2M platform

At the time of development, the OM2M platform was in Beta release as version 0.8.0, which did not implemented the full SmartM2M specifications. Setup and deployment of the NSCL were found straight forward and only required installing the Java Runtime Environment (JRE 7), configuring the host ports in a configuration file, and executing the binaries.

Overall the NSCL software proved adequate as used in System A. There were however two incidents: (1) after two months of continuous running the NSCL crashed unexpectedly with no error logs, (2) the web admin interface provided by the NSCL to inspect the resource tree became sluggish and slow to respond. Other than this, OM2M is advocated as an appropriate platform for standard compliant M2M application development. As of 22 June 2016, the *SmartM2M* standards were no longer supported by OM2M, which now only support the more recent and updated *oneM2M* standards.

4.3.3 Evaluation of the MQTT protocol and Mosquitto broker

The MQTT protocol proved highly usable and effective for EWH measurement and control. The *path* design of message topics, combined with a wildcard notation, makes configuration of data subscription simple. For example, the DTL easily subscribes to *all* EWH activity, using the "#" wildcards. The use of topics also ensures network anonymity amongst the clients.

The Paho MQTT embedded C libraries were straightforward to incorporate in the communication stack. The libraries are written to not assume any specific underlying communication, which means that they could be incorporated without any modification to fit on top of the custom AT command TCP layer. Furthermore, MQTT has a rich ecosystem of on-line community support making development and error-finding easier than that of the OM2M platform.

System B was load tested using simulated EWH clients. A maximum load of 9800 EWH clients was achieved before the DTL started dropping inbound notifications. This means that the requirement of 450 client was met with enough capacity to grow twenty times indicating good scalability.

The Mosquitto broker is easy to setup and deploy. It may be installed using the Linux *apt* package manager.

4.3.4 Evaluation of AT command interface

In Section 2.4.4.1 the three modes of AT command interfaces were introduced, of which two have been used in this work; (1) the *transparent character pipe mode* for System A, and (2) the *opaque AT command* mode for System B. The AT command interface proved to be a key challenge in both systems.

From a network architecture perspective, both the *transparent* an *opaque* AT command modes essentially replace the *data-link*, *network*, *transport* and *session* layers of the OSI network model (refer back to Figure 2.9 on page 30). In other words, the standard protocols usually found at these layers are replaced with vendor specific AT commands. This is a major concern for two reasons: (1) it decreases design flexibility because only preprogrammed AT commands can be employed to configure, use and debug communications, and (2) the software becomes dependent on the vendor, meaning that changing

vendor requires re-implementation of the AT command interface on the communication stack.

Implementation of the AT command interface was also made difficult due to lack of support. Since each modem vendor's AT command set is unique, no open-source libraries could be found that were suitable for the LEON-G100. The AT command interface also lacks community support in terms of best practices, which lead to many trail-and-error iterations of the implementation.

Comparing the *opaque* AT command mode with the *transparent* mode, it was found that although the *opaque* modem is more complex to implement, it provides greater flexibility and reliability. Since the PDP session, in *opaque* mode, is managed by the modem in the background, it allows performing AT commands unrelated to any ongoing PDP sessions. For example, a USSD code can be dialled to request the airtime balance from the provider without interrupting the data connection. In the event that simultaneous external events occur (such as receiving a TCP packet whilst making a phone call) the corresponding URCs are simply queued and handled by the AT layer. Transparent mode effectively removes all complexity from the microcontroller, but at the cost of configuration and features. The benefit of using *opaque* AT command mode is therefore that it enables sufficient functionality, with only a moderate level of complexity at the micro-controller.

Chapter 5

EWH efficiency and model validation experiment design

To effectively employ a bidirectional EWH communication system for demand-side management, an accurate model is required to predict the behaviour of each EWH (to ensure customer satisfaction). This chapter presents the experimental setup for the investigation of Objectives 2 and 3, that is, to validate the models presented in [26], and to evaluate the difference in energy efficiency between thermostat and schedule control. Section 5.1 presents the experimental methodology and metrics of interest. In light of this, Section 5.3 describes the physical setup of the lab experiment along with its expected capabilities and limitations.

5.1 Methodology

This section describes the experimental methodology used to derive the requirements for the physical setup. To evaluate Objectives 2 and 3, some laboratory experimentation using EWHs is required. For the purposes of this study *"experimentation on an EWH"* refers to installing the EWH according to SABS standards [114] in a laboratory environment and performing controlled draw-off and heating in a fashion similar to that of regular domestic use. Under no circumstances will the EWH be subjected to conditions other than its intended use. The only difference to regular domestic use is that all draw-off and heating should occur precisely and consistently.

The basic principle in EWH model validation is to perform laboratory experiments on EWHs, identify and measure all metrics of interest, then reproduce the experiments using software simulation with model, and finally, compare the results yielded.

The basic principle in comparing thermostat and schedule control schemes is to perform laboratory experiments on EWHs where both schemes are applied, identify and measure all metrics of interest, and finally compare the difference in results for the two schemes.

Both of the above cases require that consistent draw-off and heating be performed while all metrics of interest are measured. It was concluded that a single set of experimentation data can be used to evaluate both of the above enquiries. It is argued that the experimental data generated for comparing thermostat and schedule control is a sufficient

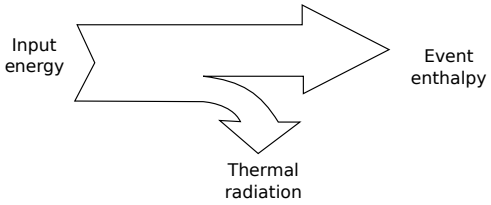


Figure 5.1: EWH model energy breakdown

dataset with which to also validate the models, since the draw-off and heating required for this experiment is representative of the type of conditions the models are expected to model. In fact, it will be interesting to investigate the differences in model accuracy under each of these control schemes.

5.1.1 Metrics

This section defines the metrics of interest for both Objectives 2 and 3. Since the EWH model presented in [26] is an *energy* model, the key metrics to measure (to evaluate the performance of the model) is the input and output energy. These are determined as follows:

Input energy: Simply determined by measuring the power supplied to the element, and integrating over the time as given in Equation 5.1.

$$Q_{\text{elec}}(t)|_{t_0}^{t_N} = \int_{t_0}^{t_N} P(t)\delta t \tag{5.1}$$

To measure the power supply to the EWH implies measuring both the supply voltage and current and then calculating the power using $P = V_{\text{RMS}} I_{\text{RMS}}$. This is possible since the EWH element is a purely resistive load and therefore has unity power factor.

Output energy: It was shown in Section 2.3.4 that energy leaves the EWH system by means of two mechanisms: (1) the enthalpy of the water drawn off and (2) thermal radiation of the cylinder boundary and pipes. The formula for calculating the delta draw-off enthalpy, Q_{draw}, was given in Equation 2.2 (page 24) and is repeated here:

$$Q_{\text{draw}} = c\rho V_{\text{draw}}(T_{\text{cyl}} - T_{\text{in}}) \tag{5.2}$$

From this, it is clear that three metrics are required, (1) water outlet temperature, (2) water inlet temperature, and (3) drawn-off volume.

The output energy due to thermal radiation (heat loss) is not practically measurable. It is assumed to be the difference between the input energy and the event output energy as shown in Figure 5.1.

Heat loss is therefore defined as:

$$Q_{\text{loss}} \triangleq Q_{\text{elec}} - Q_{\text{draw}} \tag{5.3}$$

However, Equation 5.3 assumes that the loss is calculated *over all time*. In order to calculate the losses over a *discrete* period in time, t_0 to t_N, the initial and final conditions of the enthalpy contained within the EWH needs to be taken into account:

$$Q_{\text{loss}}(t)|_{t_0}^{t_N} = U_{\text{cyl}}(t_0) - U_{\text{cyl}}(t_N) + Q_{\text{elec}}(t)|_{t_0}^{t_N} - Q_{\text{draw}}(t)|_{t_0}^{t_N} \tag{5.4}$$

Where $U_{\text{cyl}}(t)$ is the enthalpy of the water present in the cylinder at any time t as was shown in Figure 2.6 (page 25).

In order to calculate the thermal resistance, R, which is responsible for the thermal losses, Newton's law of cooling (Eq. 2.3) will be employed and requires a measurement of the ambient temperature of the EWH installation.

With thermal losses defined, a secondary metric is defined as a measure of the EWH efficiency:

$$\eta \triangleq \frac{Q_{\text{draw}}}{Q_{\text{elec}}} \tag{5.5}$$

Similarly, Equation 5.5 assumes that Q_{draw} and Q_{elec} are calculated over all time. To calculate the efficiency over a distinct period in time, t_0 to t_N the initial and final conditions need to be taken into account as follows:

$$\eta|_{t_0}^{t_N} = \frac{U_{\text{cyl}}(t_N) - U_{\text{cyl}}(t_0) + Q_{\text{draw}}(t)|_{t_0}^{t_N}}{Q_{\text{elec}}(t)|_{t_0}^{t_N}} \tag{5.6}$$

Since all the above metrics require integration over time, measurements need to be taken periodically to build a time series over which the integration can be performed. In a preliminary study it was determined that a sampling period of one second is adequate for both measurements and precise draw-off control.

5.1.2 Experiment parameters and constraints

In this section, the various experimental parameters, and how these were determined using the constraints identified, are discussed. Three experimental parameters need to be determined; (1) the pattern of water draw-off, (2) the schedule of heating to match the draw-off, and (3) the experiment duration.

Type of draw-off: It was decided that water draw-off should be a *periodic* routine over 24 hours. This decision is motivated for two reasons. Firstly, this is representative of how an EWH is generally used in a household environment [26]. Secondly, with periodic usage, it will be assumed that the internal enthalpy of the EWH (U_{cyl}) is **also** periodic. As was seen in Equations 5.4 and 5.6, calculating the heat-loss (Q_{loss}) and efficiency (η) requires knowing the initial and final enthalpy, $U_{\text{cyl}}(t_0)$ and $U_{\text{cyl}}(t_N)$ inside that cylinder. Obtaining these values, requires knowing the temperature of the water at every point inside the EWH which is not feasibly obtainable. However, if periodicity of the internal enthalpy is assumed, it implies that:

$$U_{\text{cyl}}(t) = U_{\text{cyl}}(t+T) \tag{5.7}$$

where T is the period of 24 hours. This means that if the heat loss (Q_{loss}) and efficiency (η) is calculated over a *whole* number multiple of periods, the initial and final internal enthalpy of the EWH matches and *cancel* in Equations 5.4 and 5.6.

Having motivated that the draw-off should be a daily periodic routine, the next step is to determine the specific times and volumes for draw-off suitable for the experimental goal. This task proves to be complicated since there are few constraints to guide the selection.

Draw-off event size constraints: It was decided that the *largest* single event may not be larger than 70 litres, which is 70% of the total EWH capacity. The rationale is that if larger events are required, then a larger EWH would have been installed to ensure user comfort. Since the flow meter of the Mk3 hardware can only measure to a resolution of 0.5 litres, the smallest event may not be smaller than 2 litres to ensure a tolerable event size percentage error.

Draw-off event time constraints: It was decided that the minimum timing between events may not be less than 15 minutes since this may well be seen as a single event.

No other constraints other than the ones listed above could be identified on reasonable grounds for selecting a draw-off routine. Since it cannot be known *a priori* exactly how the choice of draw-off routine might affect the results, it was decided to choose a routine that was based simply on previous experience. The assumption is that if the routine is applied *consistently* for all tests, the results should be *comparable*. Afterwards, looking at the results it may be attempted to draw some conclusions on how this particular draw-off routine affected the experiments and make recommendations for future experiments.

The chosen routine is shown by the blue bars in Figure 5.2. There are 7 events of various sizes occurring between 9 PM at night and 3 AM at the times indicated on the x-axis. The sizes are: 2 large events ($V_{\text{draw}} \geq 50\%V_{\text{cyl}}$), 2 medium events ($50\%V_{\text{cyl}} < V_{\text{draw}} \leq 10\%V_{\text{cyl}}$) and 3 small events ($V_{\text{draw}} < 10\%V_{\text{cyl}}$). It is assumed that the sum of the event volumes over a 24h period may be more than V_{cyl}, as long as some heating is done in between.

Heating schedule: The heating schedule and the relationship between the draw-off routing and the heating schedule is by definition a critical factor in determining the *efficiency difference* between thermostat and scheduled control.

Thermostat control (TC) is defined as: The mechanical thermostat of the EWH is set to its maximum value ($\pm 70°C$) and always supplied with power. The element of the EWH is therefore switched according to the thermostat, and means that the water will *always* be heated whenever the mechanical thermostat detects an internal temperature lower than the setpoint. This is the standard mode of operation for most household EWHs.

Schedule control (SC) is defined as: The mechanical thermostat of the EWH is set to its maximum value ($\pm 70°C$), but only supplied with power during certain predetermined time-slots of the day.

In order to make the results of the thermostat and schedule control comparable, the heating schedule must *match* the draw-off routine, that is, the same equivalent enthalpy must be delivered in its events as the thermostat controlled experiment. As with the draw-off routine design above, it cannot be known *a priori* what the optimal heating

$\sum V_{\text{draw}}$ (litres)	T_{in} ($^\circ C$)	T_{out} ($^\circ C$)	Q_{draw} (kWh)	Heating time (hours)
161	20	50	5.6	2.8
161	20	63	8.0	4.0
161	20	70	9.3	4.6

Table 5.1: Naive estimation of the minimum total heating time of heating schedule using Eq. 5.1 and 5.2 assuming $Q_{\text{draw}} = Q_{\text{elec}}$ and $P_{\text{elec}} = 2$ kW.

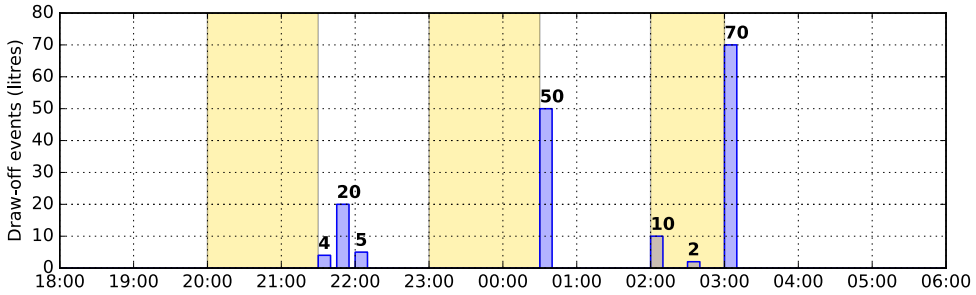


Figure 5.2: Daily water draw-off routine (blue) and heating schedule (yellow). Total draw-off volume is 161 litres. Total heating time is 4 hours.

schedule ought to be that will deliver the same drawn-off enthalpy but that requires less input energy. Again the assumption is that as long as the chosen heating schedule is applied consistently for all experiments, the results should be comparable. The selected heating schedule shown by the yellow shading in Figure 5.2. The following preliminary calculations and thumb rules were used to guide the design of the heating schedule.

A key requirement was that the customer satisfaction ought to be met at every event. That is, the temperatures of all events should be above $49^\circ C$. To minimise the effect of standing losses, water should be heated *just in time* for use. However, determining the amount of time needed to heat before each event is not possible since it requires prior knowledge of the initial temperature before each event. Since the element power is known as well as the desired output temperature and volume of each event, it is possible to determine at least the *minimum* total heating time required by balancing the input and output energies. From Figure 5.2 it may be seen that a total heating time of 4 hours per day has been selected. Table 5.1 shows a the results for calculating the total expected enthalpy at a minimum temperature of $50^\circ C$ and maximum of $70^\circ C$. It is shown that 4 hours of heating at 2 kW, falls between these two points and validates it as a reasonable selection. It is stated again that it will not necessarily ensure that user satisfaction is always met for each event, since the calculations below do not take the *timing* into account.

Experiment duration: Each draw-off test is carried out over a period of 12 days. It was determined in a feasibility study that 3 days are sufficient to reach periodic steady-state. Periodic steady-state allows the calculation of the heat-loss (Eq. 5.4) and efficiency (Eq. 5.6) since the internal enthalpy is *also* assumed to be periodic (subsequently cancelling in the calculation). This then leaves 9 days to collect enough steady-state data to make conclusions that are statistically meaningful.

To minimise the risk of interference that daily civil activity might have (such as water supply maintenance, power maintenance or general water usage), the daily draw-off routine was selected to occur during the evening between 9 PM at night and 3 AM in the morning.

5.1.3 Experimental procedure

Having defined a draw-off routine and heating schedule, the experimental procedure for determining control-type efficiency, and model validation can be now also be defined. The procedural steps are listed below followed by a discussion.

1. The draw-off routine is performed for 12 days on *two* EWHs simultaneously (one mounted vertically and the other horizontally), both EWHs under *thermostat control (TC)*
2. The procedure is repeated for another 12 days, except that that both EWHs are switched to *schedule control (SC)*
3. The EWHs are then swapped with one another and (1) and (2) are repeated, again 12 days each.

Thus in total, four, twelve day, experimental procedures will be performed on two EWHs resulting in eight data sets. It was decided that draw-off tests should be performed on two EWHs *simultaneously*; one vertically and the other horizontally orientated. This ensures that *inlet* and *ambient* temperatures are the same for at least two datasets and will not skew the difference in performance between the vertical and horizontal orientation. Testing more than two EWH at a time is unfeasible since the supply water pressure cannot support the required flow-rate. In [26] it was made clear that the orientation is an important parameter in the EWH model.

The purpose of interchanging the EWH as stated in point (3), is to ensure that any physical differences between the two EWHs (even though there ought not to be since they are the same make) and any difference the plumbing might have, can be accounted for. This also increases data variety needed to make statistically sound conclusions.

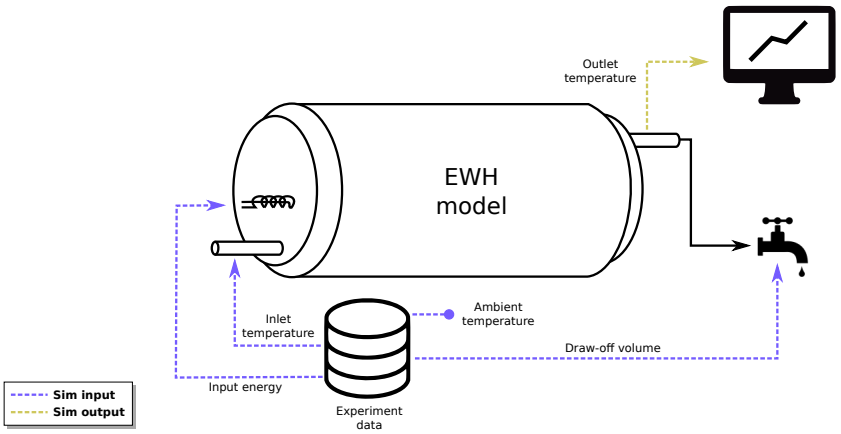
The original position of the EWHs will be referred to as **Position A**, and the latter as **Position B**. To keep track of the of the individual EWH cylinders, they will be denoted **Cylinder X**, and **Cylinder Y**. This will be relevant in the next chapter.

5.2 Model and simulation software implementation

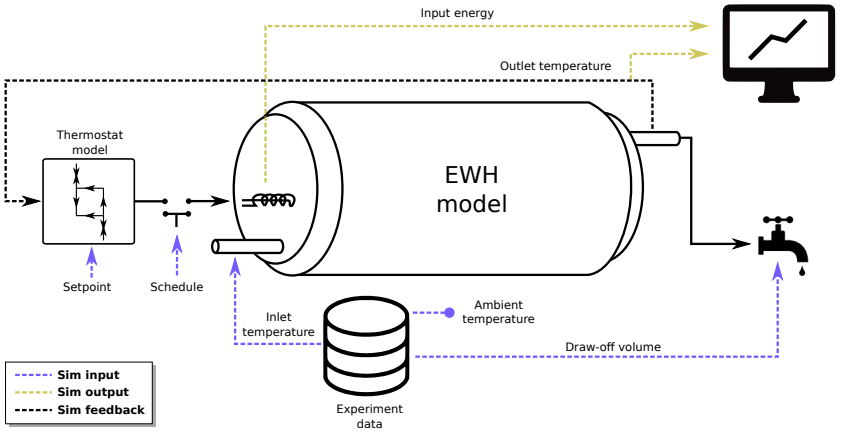
Both the one-node and two-node EWH models from Section 2.3.4 were implemented as classes in the Python programming language. All simulations, calculations and data handling were also performed in Python using the *Jupyter Notebook*. The Jupyter Notebook was chosen for its ease of use as a flexible scripting platform.

5.2.1 Two simulation approaches

EWH simulations are performed by feeding the time series data recorded using the experimental setup above, into the EWH models and extracting the simulated output. This may be done in two different ways depending on whether or not the modelling of the *thermostat* action is included in the simulation. The two approaches are defined as follows:



(a) Open-loop simulation



(b) Closed-loop simulation

Figure 5.3: Two different simulation approaches showing the data that is fed to the EWH model, and that which is extracted.

Open-loop simulation: Figure 5.3a shows the data input for the open-loop simulation. In this case the thermostat is **not** modelled. The exact element input power (P_{elec}) that was recorded during the experiment is fed into the model along with the exact measured draw-off event volumes (V_{draw}). Leaving out the thermostat eliminates any converging or diverging feedback effects the simulated thermostat may have on the simulation and isolates all behaviour strictly to the thermal aspects of the EWH model.

Closed-loop simulation: Figure 5.3b shows the data input for the closed-loop simulation. In this case the thermostat is also modelled and used as **feedback** in switching the element state using the simulated temperature as reference. That is, instead of feeding the raw element power measurements directly into the model (as with the open-loop simulation) the state of the element at any point in time is determined by the simulated temperature of the EWH at that time. For power input, the simulation only specifies a time schedule and setpoint temperature.

In both of the above cases the measured inlet water temperature (T_{in}) and ambient temperature (T_{amb}) are fed to the model.

1	**EWH installation**
1.a	Must be installed according to SABS standard 10254 [114].
1.b	Two EWHs must be installed identically as far as possible.
1.c	One EWH orientated vertically, and the other horizontally.
1.d	The thermostat on each EWH must remain in tact and set to its maximum value.
1.e	Both EWH must be supplied from the same water supply.
2	**Metrics:**
2.a	Input power in kWatt (P_{elec})
2.b	Outlet water volume (V_{draw})
2.c	Outlet water temperature (T_{out})
2.d	Inlet water temperature (T_{in})
2.e	Ambient cylinder temperature (T_{amb})
2.f	All metrics must be measured on a one second period
3	**Procedure**
3.a	Automatically open and close hot water supply at specified times
3.b	Automatically switch power on/off supply to the element at specified times
3.c	Perform closed-loop control on water volume to draw-off correct volume
4	**Infrastructure**
4.a	Data-points must be given an accurate time-stamp.
4.b	Store every data-point in a safe place for easy access
4.c	Must be robust against power failures

Table 5.2: Experimental setup derived requirements

5.3 Experimental physical layout

This section describes the physical setup of the EWH experiment, according to the methodology derived in the previous sections. The requirements of the physical setup are derived and presented in Table 5.2. The rest of this discussion refers to Table 5.2 to indicate how the requirements are met.

5.3.1 Layout

Figure 5.4a depicts the layout of the experimental setup. Only the horizontal EWH is depicted in the figure. The setup was duplicated on an adjacent vertical EWH. A photo of the full setup is shown in Figure 5.4b. A discussion follows below:

EWH cylinders: Both EWHs are installed according to SABS 10254 specification. This entails installing (1) a pressure reducing and expansion-relieve valve at the inlet, (2) an over temperature and pressure safety valve (TP valve), and (3) a vacuum breaker at the inlet and outlet with more than 30 cm head (1.a). The EWHs are wall-mounted next to each other. Both EWHs are fed from the same water supply (1.a, 1.d). Both EWHs are installed with the outlet at the top, and inlet at the bottom (1.a, 1.c). The original thermostat of each EWH is left in place and turned to maximum temperature (1.c). Both EWHs are 100 litre *Kwikot* EWHs [115] and rated to a static pressure of 400 kPa. Since the EWHs are the same make, swapping from Position A to B required *no* plumbing alterations. The inlet, outlet and safety release connections can simply be unscrewed and the cylinders swapped. This should insure that no other plumbing related variables are introduced.

Measurements: It was decided to use the Mk3 hardware by Brown [25] (described in Section 2.4) as the experimentation instrumentation for recording EWH measurements and controlling the energy supply to the EWH. The hardware is able to make the required temperature measurements (2c-e) by using analogue temperature probes. The hardware is also equipped with a relay to switch the power supply (2.b). The hall-effect flow-meter is used to measure draw-off volumes.

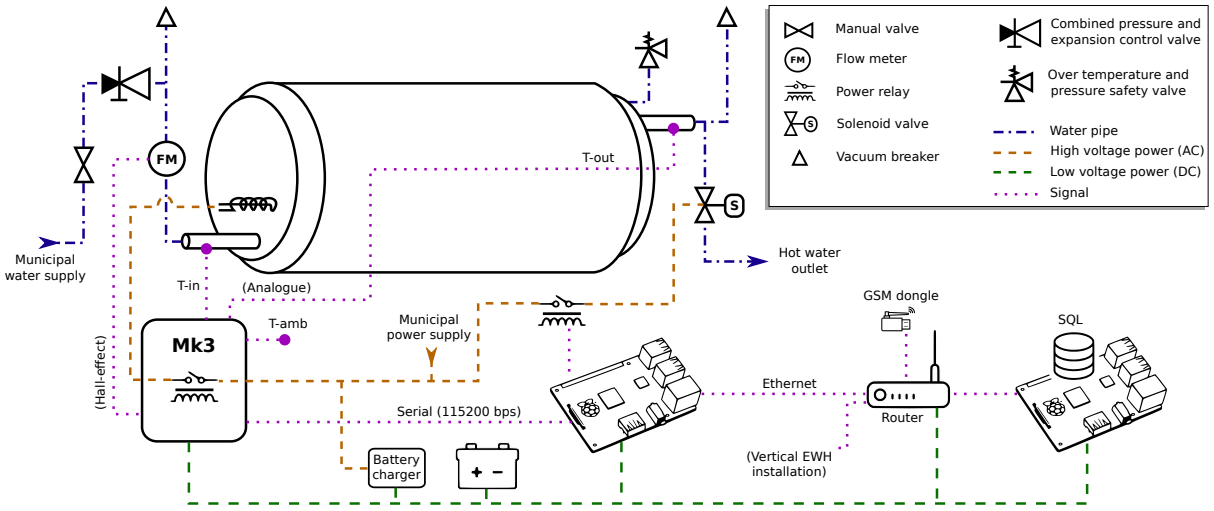
Metrics: The Mk3 hardware alone is however not sufficient to satisfy all the requirements of the experiment. For power measurement, the hardware only measures the supply *current* and assumes the voltage. To correct for this the power measurement requires calibration. The analogue temperature sensors also require calibration *at their working points* since they are not perfectly linear. The default sampling period of one minute needs to be adjusted to one second (2.f).

Data storage and control automation: To increase robustness, it was also decided to remove the modem communication dependency and perform all control and data storage locally. For this task Raspberry Pi's (RPi's) were used. As shown in Figure 5.4a each Mk3 unit is connected to a RPi via a serial line. The firmware on the Mk3 unit was modified to write all time stamps in JSON format out the serial line. The RPi parses each inbound data stamp and stores the data on a third RPi running a MySQL database server. The RPi also adds an accurate time stamp to each data point since the Mk3 real-time clock was found to drift significantly (4.a). The serial line is also used by the RPi to send commands to the Mk3 unit. Specifically to switch the element on or off (3.b).

High temperature solenoid valves are used to automatically open and close the EWH outlet in order to perform automated water draw-off (3.a). The valves are switched using a relay switching board, which is connected to the GPIO pins of the Raspberry Pi's.

All storage and control functions on the RPi are executed by a program control loop written in Python. Apart from parsing inbound data points, the control loop also listens for inbound draw-off and heating commands using a local TCP listener thread. If a *heating* command is received the control loop forwards the corresponding element state to the Mk3 unit. If a *draw-off* commands is received, the solenoid is opened and the inbound data points from the Mk3 unit are monitored until the requested volume has been drawn-off after which the solenoid is closed (3.c) (a safety timeout of maximum 30 minutes is also employed). The scheduling of draw-off and heating (as depicted in Figure 5.2) is done by specifying *cron-jobs* on the RPi's which are then periodically sent via TCP to the python control loop. Using cron-jobs makes is easy to change between TC and SC, and also keep the business logic and the control logic separate. Python was chosen, for its ease of use and its well supported libraries for serial and SQL application.

IT infrastructure Raspberry Pi's were chosen for their low power consumption, reliability, ample computational capabilities and access to accurate network time (4.a). The three RPi's are connected to a router to create local area network. The three RPi's, two Mk3 controllers and router are connected to a 107 Ah battery which is permanently connected to a trickle charger. This setup buffers the monitoring equipment from any intermittent power supply cuts during the day (4.c). Furthermore, a GSM dongle plugged into the router is used as a gateway to the Internet and ensures the RPi's have accurate network time (4.a).



(a) Lab experiment setup (refer to [114] for SABS EWH installation standards).

(b) Photo of lab setup showing both vertical and horizontal orientations. This photo was taken in Position B. Cylinders X and Y have been marked as shown.

Figure 5.4: EWH experiment laboratory installation

Chapter 6

EWH laboratory experiments and simulation results

In this chapter, the experimental results are presented for the evaluation of Objectives 2 and 3 using the methodology and experimental setup presented in Chapter 5. Firstly, the results of some additional experimentation, to determine model parameters needed for simulation, are described. Specific focus is placed on determining the thermal resistance. Afterwards, the experimental and simulation results are presented and discussed to evaluate model validity and EWH efficiency.

6.1 Model parameters

In order to evaluate EWH model accuracy, it is imperative that all model parameters be known. Specifically, these are (1) element power P_{elec}, (2) thermostat setpoint and hysteresis, and (3) thermal resistance, R. This section describes the procurement of these values that are used in the next section for simulation.

6.1.1 Element power

As stated in Section 2.4, the current supplied to the EWH element is measured by an inductive current transformer on the EWH hardware. The supply voltage is not measured by the hardware but assumed to be 230 V_{RMS}. To ensure that the power measurement is accurate, a bench digital power meter (Yokogama 2533) was used to measure each element, and a calibration factor was added to the measurements. Table 6.1 summarises the element measurements and calibration factors.

6.1.2 Thermostat set-point and hysteresis

The hysteresis of each thermostat (X and Y) was determined by using a simple bench experiment. Each thermostat was placed in a beaker with a calibrated glass thermometer. The water in both beakers were, heated and cooled five times for consistency, and the temperature points at which they switched on and off noted. Table 6.1 shows the measured values. The set-point is assumed to be the average between the hysteresis.

	X	Y	Unit
On click temperature	58.3	63.5	°C
Off click temperature	68.3	71	°C
Hysteresis	5	3.75	°C
Setpoint	63.3	67.25	°C
Element power	1.898	1.898	kWatt
Power calibration	0.9281	0.9169	N/A

Table 6.1: Summary of cylinder parameters of both EWHs

6.1.3 Thermal resistance

The thermal resistance, R, is a critical parameter for this experiment, since it is the key parameter that determines thermal losses. Two different techniques were used to determine R: (1) a static cool-down test, and (2) a static heating test. The term *static* implies that no water draw-off occurs. Before proceeding to these tests, a short discussion on the *outlet temperature accuracy* is needed.

It will be shown (in the two static experiments below) that the T_{cyl} temperature forms the foundation of the thermal resistance calculation. Since this temperature is not measured directly (as this would be invasive), it needs to be inferred. T_{out} is used to infer the internal temperature. It was found that when there was no water draw-off, the outlet temperature reading drops significantly due to the ambient effects on the external copper pipe. To compensate for this, a simple transfer function was derived to estimate the internal temperature during *static* conditions. The derivation is as follows:

Assuming the temperature difference between T_{cyl} and T_{out} is due to ambient affects on the copper pipe, the following simple linear model for estimation is proposed:

$$T_{\text{cyl}} = (T_{\text{out}} - T_{\text{amb}})\beta + \alpha \tag{6.1}$$

That is: The real internal temperature is proportional to the difference between the measured outlet temperature and the ambient temperature, and with some constant offset. This model is depicted in Figure 6.1. The values for β and α were determined by measuring the static steady-state temperatures, as well as the temperatures during water draw-off at both the upper and lower temperature range, and solving Equation 6.1. Table 6.2 summarises the measured values as well as the values calculated β and α. Table 6.2 also inherently proves the need for the transfer function since it shows a discrepancy between measured static and dynamic temperatures.

Bear in mind that this transfer function is *only* applicable during *static conditions*. It cannot be used during water draw-off since the sensors have been calibrated for water draw-off conditions as discussed in Section 2.4.

6.1.3.1 Static cooldown test:

The principle behind this test is to determine R using the *impulse-response* of the EWH system. The EWH is heated to maximum (setpoint) temperature, and allowed to cool for five days. Figure 6.2a shows the temperatures of the EWHs during cooldown (both the measured and transferred temperatures are given). The raw temperature plots for the cooldown tests may be found in Appendix G. From these temperatures R may be calculated using the Newtonian cooling law (Equation 2.3) as follows:

The first order, ordinary differential equation may be solved as:

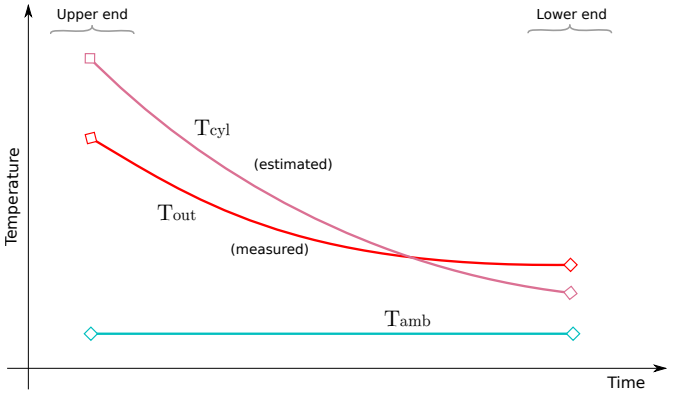
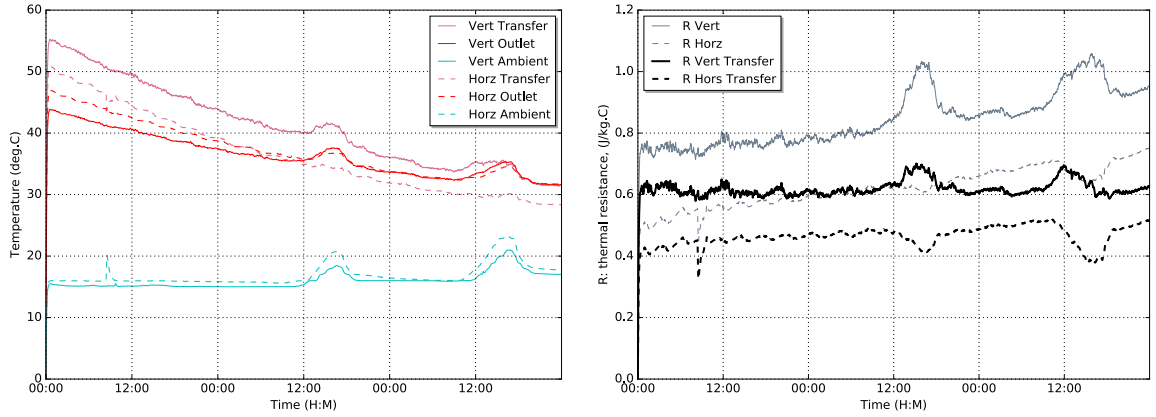


Figure 6.1: Static internal temperature estimation transfer function visualisation.

		Vert	Horz
Lower-end	T_{out} (static)	30	30
	T_{out} (draw-off)	32	33
	T_{cyl} (real)	26	25
Upper-end	T_{out} (static)	54	56
	T_{out} (draw-off)	66	60
	T_{cyl} (real)	70	62
	T_{amb}	20	20
	β	**1.83**	**1.42**
	α	**7.66**	**10.77**

Table 6.2: Static transfer function calculations.



(a) T_{out} and T_{cyl} cooling over 3 days. T_{cyl} is calculated from T_{out} using transfer function Eq. 6.1.

(b) Thermal resistances calculated over 3 days using Eq. 6.3. (both measured and transferred).

Figure 6.2: Cooldown results and calculations of EWHs in Position A.

$$T_{\text{cyl}}(t) = T_{\text{amb}}(t) - (T_{\text{cyl}}(0) - T_{\text{amb}}(t))e^{-kt} \tag{6.2}$$

Where $T_{\text{cyl}}(0)$ is the initial temperature of the EWH cylinder. Next, substituting k for R using Equation 2.5 and making R the subject, the thermal resistance can be expressed as a continuous function in time:

$$R(t) = -\frac{1}{c\rho V_{\text{cyl}}}\frac{1}{t} \times \ln\left(\frac{T_{\text{cyl}}(t) - T_{\text{amb}}(t)}{T_{\text{cyl}}(0) - T_{\text{amb}}(t)}\right) \tag{6.3}$$

By expressing Equation 6.3 as a function of time, R can be calculated over a period and *averaged* to marginalise noise and ambient effects that can be seen in Figure 6.2a. The result of the continuous time calculation is shown in Figure 6.2b. Note in Figure 6.2b, that the calculation is performed for *both* the *measured* and *transferred* temperature values from Figure 6.2a. The transferred calculation of R, yields a relatively, constant value over time, which is expected since R is dependent on the physical characteristics of the EWH cylinder and ought not change over time.

Index	A	B	C	D	E
1	Experiment info	Dataset	A	B	
2		Kill power	07 Jul 06:30	02 Sept 06:30	
3		**Initial**	07 Jul 12:30	02 Sept 12:30	
4		**Start**	08 Jul 00:00	03 Sept 00:00	
5		**End**	11 Jul 00:00	06 Spet 00:00	
6		Duration (days)	3	3	
7		EWH orientation	Horz	Vert	**Y**
8			Vert	Horz	**X**
9	T (**Initial**)	T_{out} (measured)	54.0	52.0	Y
10			48.0	53.0	X
11		T_{cyl} (transferred)	60.5	66.1	Y
12			62.4	57.7	X
13	T (**Start**)	T_{out} (measured)	47.1	48.0	Y
14			43.9	47.0	X
15		T_{cyl} (transferred)	51.0	59.2	Y
16			55.2	49.3	X
17	T(**End**)	T_{out} (measured)	31.6	34.6	Y
18			31.6	33.7	X
19		T_{cyl} (transferred)	28.2	34.5	Y
20			31.3	30.3	X
21	$\tilde{T}_{\text{amb}}$	(measured)	17.0	20.1	Y
22			16.0	20.2	X
23	$\tilde{\mathbf{R}}$ (°C/Watt)	T_{out} (measured)	0.6151	0.8059	Y
24			0.8393	0.6662	X
25		T_{cyl} (transferred)	**0.4619**	**0.5840**	Y
26			**0.6189**	**0.4807**	X

Table 6.3: Summary of thermal resistance calculation using *cooldown* test.

Table 6.3 summarises the numeric results of the static cooldown test. As stated in the previous section, T_{out} was used as an approximation for the internal temperature, T_{cyl}. Table 6.3 gives both measured values for T_{out} and the estimation of the internal temperature, to inspect the effect of the transfer. The final values used for R, are given in rows 25-26. A discussion on the table follows below.

It is important to note the following property of Equation 6.3:

$$\lim_{t\to 0} \frac{1}{t} \times \ln\left(\frac{T_{\text{cyl}}(t) - T_{\text{amb}}(t)}{T_{\text{cyl}}(0) - T_{\text{amb}}(t)}\right) \to \infty \times 0 \tag{6.4}$$

which leads to computational numeric errors. The value of t therefore needs to be sufficiently far from zero to yield accurate results. As shown in Table 6.3 (columns C,D rows 2,5), the calculation was started 12 hours after the last element on-time, and then calculated over a period of 3 days.

The cooldown experiment was performed in both positions A and B and therefore yielded four values of R, one for each EWH in each orientation (columns C,D rows 25,26). It is interesting to note that the calculation of the value of R yields different results for the *same* cylinder but in different orientations. This is unexpected since it does not seem intuitive that the orientation would affect the thermal resistance. Furthermore, the *vertical* orientation measurements both yield consistently higher values for R (0.601 on average) than that of the *horizontal* orientation (0.471 on average). It is also interesting to note the difference in the values calculated for R between the raw *measured* results (columns C,D rows 23,24) the corrected measurements using the transfer function (columns C,D rows 25,26). In all cases, the inclusion of the transfer function causes a *decrease* in the value of R. Since the pipe temperature is proportionally colder at higher internal temperatures, it is perceived that the rate of cooling is slower than it actually is. This effect is visualised in Figure 6.2a. From the perspective the cooling law (Eq. 2.3),

	A	B	C	D
1	Start	08 Sept 00:00		
2	End	12 Sept 00:00		
3	Duration	4 days		
4	Orientation		Vert	Horz
5	EWH		Y	X
6	T (°C)	$\tilde{T}_{\text{out}}$ (measured)	52.6	54.7
7		$\tilde{T}_{\text{cyl}}$ (transferred)	67.1	60.0
8	$\tilde{T}_{\text{amb}}$		20.6	20.3
9	$\tilde{E}_{\text{elec}}$ (kWh)		8.92	9.09
10	$\tilde{P}_{\text{elec}}$ (Watt)		92.9	94.7
11	$\tilde{\mathbf{R}}$	(measured)	0.345	0.363
12		(transferred)	**0.501**	**0.419**

Table 6.4: Summary of thermal resistance calculation using *static heating* test in Position B.

this effect results in a *higher* thermal resistance.

6.1.3.2 Static heating test:

To validate the values calculated for R, a second experiment was performed on the EWHs, where each EWH was put under *thermostat control* for 6 days, but again no water draw-off was performed. All heat leaving the EWH over the 6 day period is then assumed to be only due to thermal losses. The exact thermal losses can then be calculated as follows:

$$Q_{\text{loss}} = Q_{\text{elec}} = \int_0^N P_{\text{elec}}(t)\delta t \tag{6.5}$$

The rate at which thermal heat loss occurs was given in Equation 2.4 and is repeated below:

$$\dot{Q}_{\text{loss}}(t) = \frac{1}{R}(T_{\text{cyl}}(t) - T_{\text{amb}}(t)) \tag{6.6}$$

With $R = 1/(\rho V_{\text{cyl}} k)$ from Definition 2.5.

Now, since $\dot{Q}_{\text{loss}}$ is equal to the average electric input power, $\tilde{P}_{\text{elec}}$ over the 6 day period (which is measured), R may be calculated as follows:

$$R = \frac{\tilde{T}_{\text{cyl}} - \tilde{T}_{\text{amb}}}{\tilde{P}_{\text{elec}}} \tag{6.7}$$

Table 6.4 presents the numeric results of the static heating test. As with the cooldown test, the transfer function is used to estimate the internal EWH temperature ($\tilde{T}_{\text{cyl}}$) from $\tilde{T}_{\text{out}}$. Also, the leading 2 days have been truncated to ignore transient conditions (i.e. initial heating). The measured values of $\tilde{P}_{\text{elec}}$ are given in row 10 (≈ 93 Watt). The final values calculated for R using Equation 6.7, is shown in row 12. Comparing R to the values calculated from the cooldown test (Table 6.3), it was found that the values agree, implying they are suitable to be used in the simulations to follow. The raw temperature and power plots of the static heating test may be found in Appendix G.

6.2 Primary results

In this section, the experimental and simulation results are presented *as-is* and inspected for validity and general observations. In the next section the results are further scrutinised and refined to evaluate model validity and EWH efficiency.

The experimental procedures outlined in Section 5.1 were performed over the course of 2 months. This consisted of four 12-day experiments on two EWH (vertical and horizontal) and involved routine daily water draw-off and heating (under TC and SC). The procedures were fully automated with all heating and draw-off events occurring automatically. The only human intervention required was to start the experiments and swap the EWH cylinders from Position A to Position B. Afterwards, the collected experimental data was used to perform software simulations as described in Section 5.2.

Figures 6.3 and 6.4 on pages 72 and 73 give extracted plots (3-days) of the measured and simulated data obtained for one EWH during one of the four, 12-day experiments. Figure 6.3 shows the data obtained for an EWH under **TC**. Figure 6.3a shows the *open-loop* simulation result and Figure 6.3b the *closed-loop* simulation result (only two-node model shown). Figure 6.4 shows the data obtained for an EWH under **SC**. Again, both open-loop and closed-loop simulations are shown.

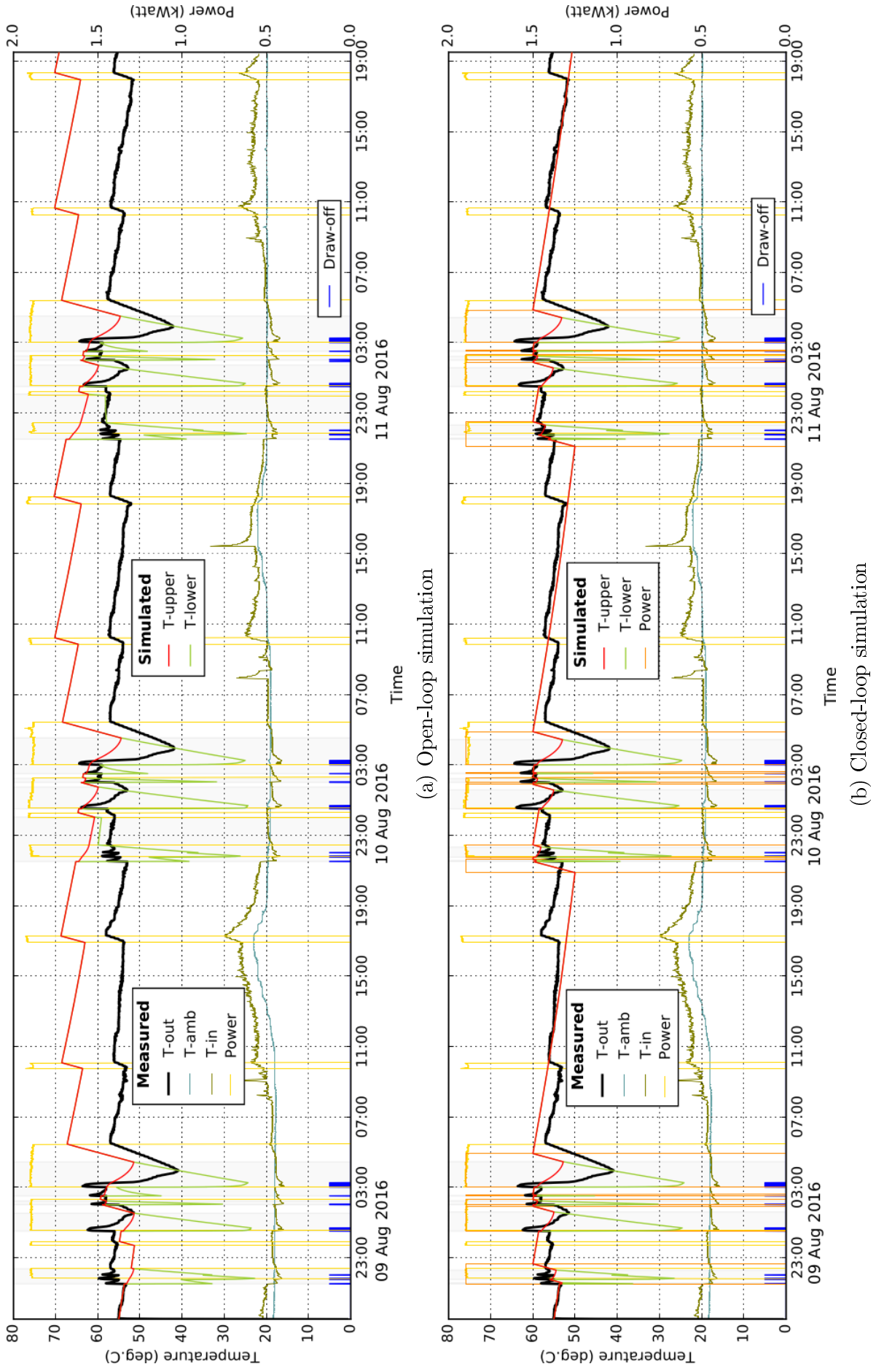
Table 6.5 presents a full summary of the *draw-off event* metrics for the complete experiment. Specific cells and ranges will be referred to by the indices on the margins of the table. The eight data-sets, given in columns D to K, are the result of four, 12-day experiments on two EWHs. Columns D to G comprises the **TC** results, and columns H to K comprises the **SC** results. The mean values of the TC and SC results are presented in columns L to M. Position A data is highlighted in yellow and Position B data in blue. Cylinder X is highlighted in red and and cylinder Y in green. The rows are as follows:

Rows (1-6) - Experiment info: Metadata on the datasets showing the date and time of execution. The eight data sets are grouped in four groups corresponding position A and B. Each dataset consists of 9 days of data at 1 second sample period. Each 9 day set is a subset of a complete 12 day experiment with the leading 3 days removed to mitigate non-periodic behaviour (explained in Section 5.1.2 on page 58).

Rows (7-14) - Physical constants and model parameters: The physical parameters including cylinder orientation and simulation power correction factor as determined in the previous section (6.1), concerning each data-set is give here. These are the parameters used for simulation.

Rows (15-20) - Control variables: The control variables were derived in the experimental design (Section 5.1.2). These are the variables meant to be kept constant across all data-sets. They are presented here to inspect their consistency.

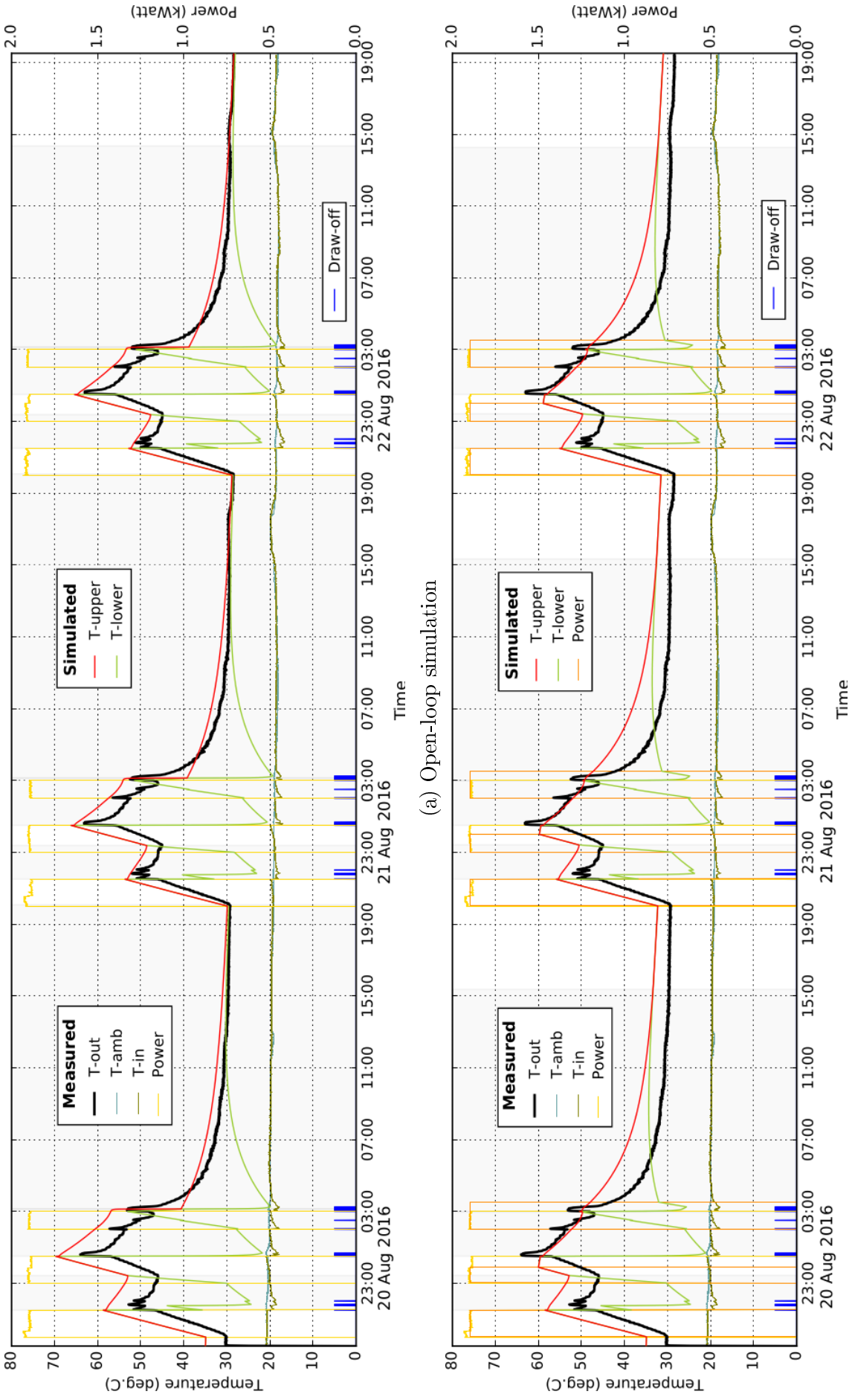
Rows (21-38) - Measured and simulated results: Presents the results that will be used to evaluate the objectives. Both the measured and simulated results are presented. The measured values are highlighted in grey in each category. For the simulations, both the one-node and two-node results are given *and* for each, the open-loop and closed-loop simulations were performed. It is important to note that the electrical and enthalpy energies are given **per 100 litres** drawn-off water. This is to normalise against the total



(a) Open-loop simulation

(b) Closed-loop simulation

Figure 6.3: Temperature and power plots of horizontally orientated EWH in position B under **thermostat control** and using two-node model for simulation. Grey shading indicates times when model in two-node state. (Cross-reference with Table 6.5, column G.)



(a) Open-loop simulation

(b) Closed-loop simulation

Figure 6.4: Temperature and power plots of horizontally orientated EWH in Position B under **schedule control** and using two-node model for simulation. Grey shading indicates times when model in two-node state. (Cross-reference with Table 6.5, column K.)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Control type		**Thermostat control (TC)**				**Schedule control (SC)**					
2													
3	Experiment info	Dataset		A		B		A		B			
4		Start		16/06/11 19:30		16/08/06 19:30		16/06/24 19:30		16/08/17 19:30			
5		End		16/06/20 19:29		16/08/15 19:29		16/07/03 19:29		16/08/26 19:29			
6		Duration		8 days, 23:59:59		8 days, 23:59:59		8 days, 23:59:59		8 days, 23:59:59			
7	Physical constants and model parameters	Orientation		Vert	Horz	Vert	Horz	Vert	Horz	Vert	Horz	Mean TC	Mean SC
8		Cylinder		X	Y	Y	X	X	Y	Y	X		
9		Element power (kWatt)		1.898	1.898	1.898	1.898	1.898	1.898	1.898	1.898		
10		Power calibration factor		0.9281	0.9169	0.9169	0.9281	0.9281	0.9169	0.9169	0.9281		
11		Thermostat setpoint ($^\circ C$)		63.35	67.25	67.25	63.35	63.35	67.25	67.25	63.35		
12		Thermostat hysteresis ($^\circ C$)		5	3.75	3.75	5	5	3.75	3.75	5		
13		T_{cyl} initial ($^\circ C$)		63.35	67.25	67.25	63.35	35	35	35	35		
14		R		0.6189	0.4619	0.584	0.4807	0.6189	0.4619	0.584	0.4807		
15	Control variables	Number of events		63	63	63	63	63	63	63	63	63	63
16		$\sum V_{\text{draw}}$ (litres)		1541.0	1462.5	1573.5	1480.5	1525.5	1481.0	1541.5	1481.0	1514.4	1507.3
17		Mean V_{draw} flowrate (litre/minute)		3.2	2.5	3.2	3.1	3.3	3.2	3.2	3.1	3.0	3.2
18		Mean V_{draw} duration (minutes)		6.5	8.4	6.5	6.1	6.2	6.1	6.3	6.1	6.9	6.2
19		$\tilde{T}_{\text{in}}$ (weighted by V_{draw})		17.63	17.09	17.77	17.21	16.09	15.71	18.31	17.80	17.4	17.0
20		$\tilde{T}_{\text{amb}}$ ($^\circ C$)		18.0	17.7	19.3	19.0	15.9	16.4	19.3	19.5	18.5	17.8
21	$\mathbf{Q}_{\text{elec}}$ (kWh per 100 litres)	Measured		7.41	8.97	7.61	6.61	4.47	4.49	4.37	4.59	7.21	4.48
22		1 Node (closed-loop)		5.35	6.35	5.87	5.72	4.44	4.56	4.39	4.56	5.82	4.49
23		2 Node (closed-loop)		6.47	7.37	6.89	6.99	4.44	4.56	4.39	4.56	6.93	4.49
24	$\mathbf{Q}_{\text{draw}}$ (kWh per 100 litres)	Measured		5.52	4.73	5.70	5.07	3.94	4.20	3.93	4.31	5.43	4.09
25		1 Node	Open-loop	5.83	6.45	5.96	4.85	3.91	3.81	3.78	3.92	5.55	3.86
26			Closed-loop	4.31	4.77	4.70	4.33	3.76	3.74	3.69	3.77	4.53	3.74
27		2 Node	Open-loop	6.25	7.00	6.40	5.26	4.22	4.08	4.11	4.19	5.97	4.15
28			Closed-loop	5.52	5.89	5.81	5.69	4.03	4.06	3.97	4.08	5.73	4.03
29	η ($\frac{Q_{\text{draw}}}{Q_{\text{elec}}}$)	Measured		74.5%	52.7%	74.9%	76.7%	88.1%	93.6%	89.9%	93.8%	75.4%	91.4%
30		1 Node	Open-loop	78.8%	71.9%	78.4%	73.3%	87.3%	84.9%	86.6%	85.5%	76.8%	86.1%
31			Closed-loop	80.6%	75.1%	79.9%	75.7%	84.8%	81.9%	84.0%	82.5%	77.8%	83.3%
32		2 Node	Open-loop	84.4%	78.0%	84.1%	79.5%	94.3%	90.9%	94.1%	91.4%	82.7%	92.7%
33			Closed-loop	85.2%	80.0%	84.4%	81.3%	90.7%	88.9%	90.4%	89.4%	82.7%	89.9%
34	$\tilde{\mathbf{T}}_{\text{out}}$ (weighted by V_{draw})	Measured		65.09	57.76	66.80	60.84	50.00	51.88	52.12	54.85	62.62	52.21
35		1 Node	Open-loop	67.83	72.58	69.07	58.93	49.70	48.50	50.86	51.54	65.28	50.15
36			Closed-loop	54.71	58.14	58.17	54.47	48.46	47.85	50.05	50.21	56.37	49.14
37		2 Node	Open-loop	71.43	77.29	72.83	62.46	52.41	50.84	53.70	53.89	68.91	52.71
38			Closed-loop	65.09	67.80	67.77	66.13	50.74	50.61	52.45	52.91	66.70	51.68

Table 6.5: Complete results table showing the average draw-off event metrics of eight datasets. Orange cells have been omitted from analysis due to experiment anomaly.

draw-off volume to give comparable values (row 16 shows that total draw-off volume varies slightly between datasets).

The reader is urged to consider the Figures 6.3 and 6.4, and Table 6.5 carefully before continuing, as they form the foundation of the discussions to follow.

6.2.1 Experimental performance evaluation

Before investigating the results, the performance of the experimental design and procedure is discussed to determine its appropriateness and reliability for the investigation to follow.

All 63 events (7 per day, for 9 days) of each dataset executed correctly and at the intended time. The system also experienced no loss of data, and was not affected by external activity. No water or power supply issues were detected. The IT infrastructure therefore proved extremely reliable and effective, especially considering the alternative of performing the experiment manually.

From the control variables (Table 6.5, rows 15-20) an anomaly as a results of the plumbing setup can be seen. Comparing columns D, F, H, J with E, G, I, K (row 16) it is seen that the total draw-off volume for the *horizontally* orientated EWHs are consistently lower than that of the *vertically* orientated EWHs (4.47% on average). This is independent of the specific cylinder since the phenomenon occurs both in Position A and B. An outlier is seen in column E, row 17 where the mean flow-rate is only 2.5 l/min compared to the average 3.1 l/min in the other data-sets. The exact cause of this phenomenon could not be established and attributed to differences in plumbing.

To compensate for the slight mismatch in draw-off volume, the electrical and enthalpy measurements were normalised against the total draw-off volume and given per 100 litres as mentioned earlier.

Comparing the measured TC efficiency (Table 6.5, columns D-G row 29), the value in column E is significantly lower than the other three. Investigating the temperature plots, it was found that the inlet temperature sensor reported unexpectedly high readings during times of no draw-off *under* TC than under SC (see T_{in} in Figure 6.3a around 10:00 and 18:00 every day). It is hypothesised as the EWH heats, some of the expanding water is exchanged via the expansion-relief valve at the inlet. This effect would cause additional energy losses not accounted for by the EWH model and would be magnified during TC since the EWH is continually heated. Again, an outlier is seen in column E row 21 where the mean electrical energy is 8.97 kWh/100l compared to the $\approx$ 7 kWh/100l in the other data-sets. Due to the irregularity of the results in column E, all *measured* and *open-loop* simulation data (in column E) are excluded from further analysis (highlighted in orange).

The mean inlet temperature (row 21) shows good consistency across the eight data-sets, at an average temperature of 17.2°C and a standard deviation of only 0.89°C. The mean ambient temperature (row 22) is also fairly consistent at an average of 18.1°C and a standard deviation of 1.38°C.

From the above, the experiment was deemed a success and acceptable to make valid conclusions from the data. Where any further discrepancies arise, it will be noted.

6.3 Model validation

This section analyses the results that were obtained through performing simulations using the one-node and two-node EWH models in an attempt to validate them (as set out in Objective 2).

The principle of the validation is to perform the same procedures described in Section 5.1.3 in simulation and compare the simulation output with actual measurements. As discussed in Section 5.2, this is done by feeding time series data, recoded during experimentation, into the model and extracting the output metrics. The output metrics that will be compared is the average *event enthalpy* ($\tilde{Q}_{\text{draw}}$), *energy consumption* ($\tilde{Q}_{\text{elec}}$), *efficiency* (η) and *event temperature* ($\tilde{T}_{\text{out}}$) (see Table 6.5 rows 24-38).

As discussed, two types of simulations are performed, namely, (1) an *open-loop* simulation, and (2) a *closed-loop* simulation. The *open-loop* simulation is used to validate the *thermal* aspect of the model, and the *closed-loop* simulation is used to validate the *operational* aspect of the model. The open-loop simulation is required since the action of the thermostat inherently stabilises the simulation loop. Closed-loop simulation is required since the open-loop simulation cannot be used for EWH state *prediction*. Open-loop simulation implicitly assumes the element state is known *beforehand*. Furthermore, both the TC and SC experimental data have been used for simulation, meaning that the model accuracy under *each* of these conditions was evaluated. In the case of the SC, *closed-loop* simulation, the schedule was also added to the simulation, meaning that the simulated thermostat could only be switched on during the same periods as the actual element (refer back to Figure 5.3b on page 62).

6.3.1 Aggregate event metrics accuracy:

This section focuses on the ability of the models to predict the accuracy of event related metrics. Table 6.6 presents a summary of the model accuracy results. For this investigation, TC and SC simulations are compared separately. For both control strategies, *three perspectives* of accuracy are evaluated: (1) one-node versus two-node model (Table 6.6 columns D-E), (2) open-loop versus closed-loop simulation (columns F-G) and (3) vertical versus horizontal orientation (columns H-I). Three metrics are evaluated, input energy (rows 1-4), efficiency (rows 5-8) and mean draw-off temperature (rows 9-12). Table 6.6 is derived from Table 6.5 where the *mean* has been calculated from each perspective. For example, the value found in (Table 6.6 cell E,11) is the mean draw-off temperature estimated by the one-node model during schedule control and is calculated from the values found in (Table 6.5 cells H-K, 35-36). In Table 6.6, rows 1-12 show the mean simulated values, along with the *real* values in column J. Rows 13-24 show the percentage error from rows 1-12 with the *average absolute* error in column J.

Input energy accuracy: As shown for TC (rows 13-14), simulations from all perspectives *underestimate* the electrical input energy required, with the one-node model the worst at 19.2% below the real value. The two-node model shows much greater accuracy (-3.8%) than the one-node (-19.2%). (Note that by definition, only the closed-loop simulation is relevant for this metric.) Simulations for SC (rows 15-16) performed all round better than that of TC with an average absolute error of only 0.8%.

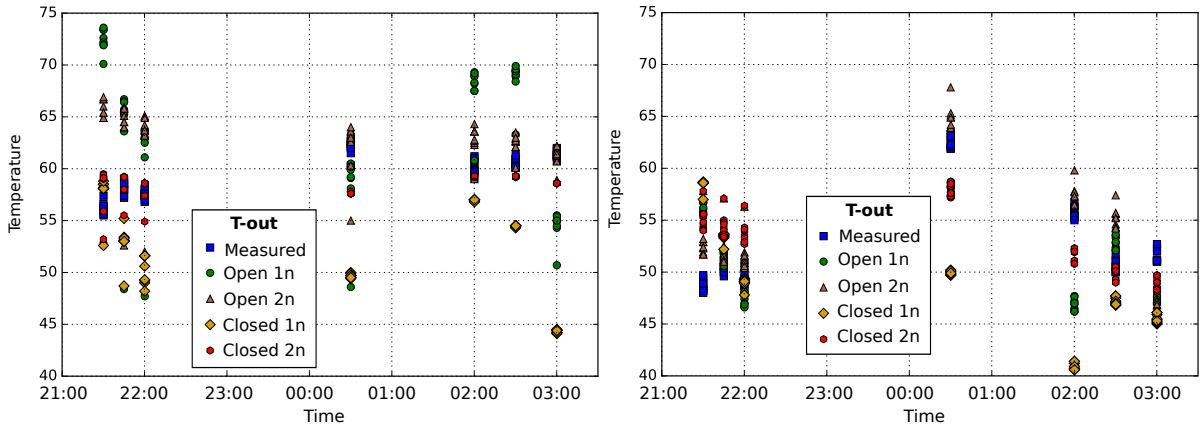
	A	B	C	D	E	F	G	H	I	J
				Perspective						
		Metric	**Control**	**Node**		**Sim type**		**Orient**		**Real**
1	Mean values per catagory	Elec per 100 litre [kWh]	TC	One	5.8	Open	N/A	Vert	6.1	**7.2**
2				Two	6.9	Closed	6.4	Horz	6.6	
3			SC	One	4.5	Open	N/A	Vert	4.4	**4.5**
4				Two	4.5	Closed	4.5	Horz	4.6	
5		Eff.	TC	One	77.4%	Open	79.6%	Vert	82.0%	**75.4%**
6				Two	82.7%	Closed	80.3%	Horz	77.5%	
7			SC	One	84.7%	Open	89.4%	Vert	89.0%	**91.4%**
8				Two	91.3%	Closed	86.6%	Horz	86.9%	
9		Mean draw-off temp [degC]	TC	One	60.2	Open	67.1	Vert	65.9	**64.2**
10				Two	67.6	Closed	61.5	Horz	61.3	
11			SC	One	49.6	Open	51.4	Vert	51.0	**52.2**
12				Two	52.2	Closed	50.4	Horz	50.8	
										|Error|
13	% error from real	Elec per 100 litre [kWh]	TC	One	-19.2%	Open	N/A	Vert	-14.7%	**11.5%** ↓
14				Two	-3.8%	Closed	-11.5%	Horz	-8.3%	
15			SC	One	0.2%	Open	N/A	Vert	-1.5%	**0.8%** ↑
16				Two	0.2%	Closed	0.2%	Horz	1.8%	
17		Eff.	TC	One	2.7%	Open	5.7%	Vert	8.8%	**6.0%** ↑
18				Two	9.8%	Closed	6.5%	Horz	2.8%	
19			SC	One	-7.3%	Open	-2.2%	Vert	-2.6%	**3.7%** ↓
20				Two	-0.1%	Closed	-5.2%	Horz	-4.9%	
21		Mean draw-off temp	TC	One	-6.3%	Open	4.4%	Vert	2.5%	**4.6%** ↓
22				Two	5.3%	Closed	-4.2%	Horz	-4.5%	
23			SC	One	-4.9%	Open	-1.5%	Vert	-2.2%	**2.5%** ↓
24				Two	0.0%	Closed	-3.4%	Horz	-2.7%	

Table 6.6: Average model accuracy results from three perspectives, split between thermostat control and schedule control (derived from Table 6.5).

Efficiency accuracy: Since the input energy for TC was underestimated, it is not surprising to find that the efficiency for TC (rows 17-18) is *overestimated* from all perspectives (refer to Eq. 5.5). SC on the other hand (rows 19-20), underestimated the efficiency even though it accurately predicted the input energy. This is because the mean-draw off temperature was underestimated all round for SC (rows 23-24). On average, the SC simulation shows better accuracy (at 3.7%) than that of TC (at 6%). The one-node model performed better at TC (2.7%) but the the two-node performed better at SC (-0.1%). No remarkable difference between open-loop and closed-loop simulation is observed (columns F,G, rows 5-8) meaning that the inclusion of simulating the thermostat feedback does not seem to have a significant effect on model behaviour. This can also be visually deduced by comparing the open-loop and closed-loop plots in Figures 6.3 and 6.4.

Finally, in the static heating tests, it was found that there was a slight difference in the value of R between the vertically and horizontally mounted EWHs. Comparing rows 5 and 7 with rows 6 and 8 (Table 6.6, column I) the calculated efficiencies concurs with the results of the static heating tests: The higher value of R for the *vertically* mounted EWH (0.60°C/W), results in a higher efficiency (for both TC and SC). The the lower value of R for the *horizontally* mounted EWH (0.47°C/W), results with a lower efficiency (for both TC and SC). The average difference is calculated at 3.3%.

Mean draw-off temperature: As mentioned, SC underestimates on the whole the mean draw-off temperature. On the other hand (on the whole) the results for TC are varying both over- and underestimating from each perspective (Table 6.6, rows 21-22). For TC, the one-node model underestimates the outlet temperature on average by 6.3% (cell E,21), whereas the two-node model over estimates the outlet temperature with roughly the same error (5.3%) (cell E,22). This makes sense since the one-node model assumes *instantaneous mixing* of inlet water, which takes no thermocline into consid-



(a) Thermostat control (dataset G). (b) Schedule control (dataset K).

Figure 6.5: Mean draw-off temperature plot per daily routine event for real and simulated values. (These events are from the horizontal EWH in Position B (cylinder X). Cross reference with Table 6.5 columns G,K and Fig. 6.3 and 6.4, and also with Fig. 5.2).

eration. The two-node model however, assumes a "hard" thermocline, which does not take any turbulent mixing into account. Again, as before, SC simulations shows better accuracy (at 2.5% absolute error) than that of TC (at 4.6% absolute error).

6.3.2 Individual event metrics accuracy:

The previous section considered the *overall average* of event accuracy. It is also necessary to inspect the individual event metrics more closely to avoid possible error oversight caused by the averaging process. Specifically the mean event draw-off temperature needs to be verified to satisfy user comfort levels.

Figure 6.5 depicts two scatter plots of the mean draw-off temperature of TC and SC experiments (from datasets G and K in Table 6.5). The 63 events from 9 days have been superimposed on 24 hours, which results in a spread along the routine draw-off time for each of the seven daily events. The actual measured temperature as well as four simulated temperatures are shown. Table 6.7 summarises the average event temperature of each daily event. The seven columns correspond to each routine event. In Table 6.7, rows 1-10 shows the average measured and simulated values, and rows 11-18 reflects the percentage error of simulation for the above corresponding rows. The final column shows the standard deviation across the events.

From these results the following observations are made: Measured values show that there is a variation in temperature for each event in the day, with the TC temperatures slightly more consistent over a day than SC (column K rows 1,5 vs rows 6,10). All events meet the user comfort constraint, but with minimal margin for error for SC control indicating a conservative heating schedule (row 6). The two-node model is significantly more accurate than the one-node model per event (compare rows 11, 13, 15, 17 with 12, 14, 16, 18). Comparing the temperature spread per event in Figures 6.5a and 6.5b, it again seems that the models show a slight affinity for SC control since it shows less spread for a particular event. However, in this case the two-node model is slightly more accurate under TC (Table 6.7 column K rows 12,14).

	A	B	C	D	E	F	G	H	I	J	K
	Daily event number			**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Std dev**
	$\dot{V}_{draw}$ [litre]			4	20	5	50	10	2	70	
	Start time			21:30	21:45	22:00	00:30	02:00	02:30	03:00	
1	**TC** (B Horz)	Measured		56	58	57	62	60	61	61	2.2
2		Open-loop	1n	70	64	61	58	68	69	54	5.8
3			2n	64	64	63	62	63	62	61	1.1
4		Closed-loop	1n	58	53	50	50	57	54	44	4.7
5			2n	58	59	58	58	59	59	59	0.6
6	**SC** (B Horz)	Measured		49	51	49	62	56	51	52	4.9
7		Open-loop	1n	56	51	47	58	47	53	48	4.5
8			2n	53	52	52	65	57	55	46	5.8
9		Closed-loop	1n	58	53	49	50	41	47	46	5.6
10			2n	55	54	54	58	52	50	49	3.2
											|Error|
11	% **TC** error	Open-loop	1n	25%	9%	6%	-6%	12%	13%	-11%	12.3%
12			2n	15%	9%	9%	0%	4%	3%	0%	5.6%
13		Closed-loop	1n	3%	-9%	-12%	-20%	-5%	-10%	-28%	9.9%
14			2n	4%	1%	1%	-7%	-2%	-2%	-5%	3.6%
15	% **SC** error	Open-loop	1n	16%	2%	-4%	-7%	-16%	4%	-7%	10.1%
16			2n	9%	4%	5%	4%	3%	9%	-10%	6.5%
17		Closed-loop	1n	20%	6%	-1%	-20%	-26%	-7%	-12%	15.7%
18			2n	13%	8%	9%	-7%	-7%	-2%	-5%	8.5%

Table 6.7: Mean draw-off temperature per routine daily event. Both measured and simulated values are presented, along with the corresponding percentage error. (These events are from the horizontal EWH in position B. Cross reference with Table 6.5 column G,K and Fig. 6.3 and 6.4, and also with Fig. 5.2)

6.3.3 Continuous time accuracy: Figures 6.3 and 6.4

Up to this point it has been implicitly assumed that the accuracy of the model is investigated from a *discrete event output* perspective. By doing so, the *continuous temporal* aspect of the simulation is implicitly ignored, meaning simulated temperature and energy metrics during times of *non draw-off* are not scrutinised for accuracy. It is argued that the EWH state during times of non-use is irrelevant. However, it is still important to assess the simulation results over all time as a sanity check on the draw-off results, and to glean additional model behaviour.

From a visual comparison of Figures 6.3 and 6.4, it is observed that the simulated outlet temperature appears to show better correlation to measured values for SC than for TC. This observation agrees with the above results that modelling under SC yields greater accuracy than under TC. Comparing the closed-loop simulations (Figures 6.3b and 6.4b) it is further observed that the simulated *heating times* under SC correlate much better with the actual heating times, than for TC. In other words, the temporal behaviour of the model is more predictable under SC. This is to be expected since a pre-defined heating schedule coerces the EWH behaviour.

6.3.4 Summary of model validation

In summary, a breakdown of the important overall result of model accuracy suggests:

- Models show better accuracy for SC than TC *independent* of orientation, simulation type and model type. That said, the accuracy (in general) seems good with the mean draw-off temperature less than 5% error at most (Table 6.6, rows 21-24).
- Overall, open-loop and closed-loop simulations yield similar simulation error with a difference in absolute error on average only 1.5% (Table 6.6 column G row 17-24).

This means that inclusion of the thermostat in the simulation did not significantly affect the thermal model behaviour. Since closed-loop simulations yield valid results, the models are useful for EWH state *prediction* purposes.

- The one-node model tends to underestimate draw-off temperature, whereas the two-node model tends to overestimate it.
- The customer comfort constraint of $T_{\text{out}} \geq 49°C$ is met during all experiments. However, since water temperature under SC is on average $10°C$ colder than under TC, it is noted that the heating schedule is substantially conservative.
- Closed-loop simulation indicates that element switching times are predicted more accurately for SC than for TC. In other words, by constraining EWH heating times to specific time-slots of the day, model predictability is enhanced.
- No remarkable difference in accuracy could be noted between vertical and horizontal model. In Table 6.6, column I, rows 13-24, the average absolute error for the vertical EWH is 5.4% and for the horizontal is 4.7%.

6.4 EWH efficiency: Thermostat versus Schedule control

In this section, the difference in energy efficiency between TC and SC is discussed to complete Objective 3. It is shown at the outset that the energy efficiencies of the TC and SC experiments *are not comparable*. Subsequently, to make the comparison, additional simulations were performed using the models that were validated in the previous section.

6.4.1 Comparing thermostat and schedule control

Referring back to Table 6.6, the reader may have noticed the difference in efficiency between TC and SC. In cell J,5, TC shows a calculated efficiency of 75.4%, whereas in cell J,7, SC shows a calculated efficiency of 91.4%. At first glance it would seem that the SC control scheme is more energy efficient than that of TC under the proposed experimental conditions (by 16.0% points). It must be noted however, that the efficiencies of the two control schemes may **not** be compared directly, since the *effective output* of the two systems are not the same. Cell J,9 shows a mean draw-off temperature of $64.2°C$ for TC, whereas cell J,11 shows a lower draw-off temperature for SC of $52.2°C$. The reason why the efficiencies may not be compared is readily illustrated when looking at an extreme case of an "always off" schedule, in which case $Q_{\text{elec}} = Q_{\text{draw}}$, and (from Definition 5.5) this would mean $\eta = 1$; that is 100% efficiency. In order to make the comparison, the same effective "work" needs to performed in both cases.

Seeing that the mean draw-off event temperature is colder for SC than for TC, it was concluded that the heating schedule selected in Section 5.1 was conservative and does not completely "match" the draw-off routine. In order to compare TC and SC, either the heating schedule needs to be adjusted for SC, or the setpoint needs to be lowered for TC. To minimise the risk of introducing additional effects due to changing the heating schedule, the latter approach was opted for.

	A	B	C	D	E	F	G	H	I
1	Experiment type	Closed-loop simulation of TC datasets							
2	Dataset	A				B			
3	Orientation	Vert	Vert	Horz	Horz	Vert	Vert	Horz	Horz
4	Cylinder	X	X	Y	Y	Y	Y	X	X
5	Model	1 Node	2 Node	1 Node	2 Node	1 Node	2 Node	1 Node	2 Node
6	Simulation set-point	**56**	**48.3**	**55**	**48.5**	**58.5**	**50.5**	**57.45**	**50.5**
7	$\tilde{Q}_{\text{elec}}$ / 100 litres	4.48	4.53	4.80	4.89	4.73	4.69	5.01	5.04
8	$\tilde{Q}_{\text{draw}}$ / 100 litres	3.60	3.85	3.57	3.91	3.75	4.01	3.84	4.15
9	η_{therm}	80.3%	84.9%	74.5%	79.9%	79.4%	85.5%	76.7%	82.4%
10	$\tilde{T}_{\text{out}}$ (weighted)	48.57	50.71	47.8	50.7	50.0	52.2	50.3	52.9
11	$\tilde{T}_{\text{out}}$ match error	0.24%	-0.04%	-0.04%	0.17%	-0.02%	-0.38%	0.14%	0.06%
12	Setpoint decrease	11.60%	23.76%	18.22%	27.88%	13.01%	24.91%	9.31%	20.28%
13	$\eta_{\text{sched}} - \eta_{\text{therm}}$	**4.48%**	**5.84%**	**7.40%**	**9.01%**	**4.62%**	**4.88%**	**5.86%**	**7.04%**

Table 6.8: Closed-loop simulation results using *TC* experimental data and adjusting the simulated setpoint until mean outlet temperature ($\tilde{T}_{\text{out}}$) matches that of the corresponding simulated *SC* experiment.

The problem can be phrased as follows: *Which temperature setting under thermostat control, will yield the same average event temperature under schedule control, given the water draw-off is the same in both cases (i.e. effective energy output)?*

It was decided to perform the investigation in *simulation* using the one-node and two-node models from the previous section. The following procedure was conducted:

The closed-loop simulation was performed multiple times on the TC datasets (Table 6.5 columns D-G) using various thermostat set-points until, in each case, the mean outlet temperature ($\tilde{T}_{\text{out}}$) matched that of the corresponding SC closed-loop simulations (Table 6.5 columns H-K rows 36,38).

Using simulation is motivated as follows:

1. It was considered unfeasible to re-perform the physical experiments iteratively since duplicating exact control conditions over a long period of time is difficult. For example, the duration would include seasonal effects. In simulation, the input data is constant for each iteration.

2. The models have been validated in the previous section to such a degree that they can be trusted to yield a trustworthy result - at least under the same conditions by which they were validated, which is the case here.

6.4.2 Results

Table 6.8 summarises the simulation results. It follows the same structure of Table 6.5. Row 6 shows the setpoints that produced the draw-off temperature match. It is important to note that the one-node and two-node simulations have been separated since it was shown in the previous section that they predicted different draw-off temperatures. Row 11 shows the temperature match error (an absolute error of less than 0.5% was deemed acceptable). Row 12 shows the percentage decrease in setpoint temperature from the thermostat experiment (Table 6.5 columns D-G row 10) that is required for a temperature match. Notice that the average setpoint decrease for the two-node model is 24.2%, whereas for the one-node it is only 13.0%. This agrees with the results in the previous section (6.3) where it was found that the two-node model over estimates the event temperature where the one-node underestimates it. Row 13 presents the difference in efficiency

between the SC simulation and the TC simulation at a lower set-point ($\eta_{\text{sched}} - \eta_{\text{therm}}$).

As before, it is noted that the two-node model predicts in each instance a slightly higher efficiency ($\approx 5\%$) than the corresponding one-node model (row 9), again due to the higher prediction in draw-off temperature (row 10). No remarkable difference between the horizontal and vertical EWH model can be observed.

The final efficiency results (row 13) are consistent and show that under the proposed experimental conditions, SC yields on average 6.1% better efficiency than that of TC given the same effective enthalpy. This confirms that the comparison made earlier between the efficiency TC and SC in Table 6.6 as 16% difference, was in fact naive and incorrect. Again it should be stated that the above conclusions have been made *only with respect to the proposed experimental conditions*. The reader is reminded that the experiment consisted of one specific draw-off routine and one specific heating schedule in conjunction with that draw-off routine. These results cannot necessarily be extrapolated to the general case and requires further investigation as to what the effects may be of any *particular* draw-off routine and heating schedule.

Chapter 7

Conclusions and Recommendations

This chapter assesses the problem statement and objectives presented in the Introduction, and concludes this study with recommendations for future work.

The use of ripple control systems to switch household EWHs is a rudimentary way for utility providers to manage peak demand. One problem with ripple control is that it cannot take individual customers into consideration, since the communication system is unidirectional and can only switch EWHs in bulk. Ideally, a system is required that not only monitors and controls the current state of each EWH, but also predicts the future state of each EWH to enable efficient peak demand management.

This study presented the development of two separate, but complementary parts of a system as an alternative to current ripple control systems.

Firstly, two communication systems were developed as potential candidates that may be used to establish a large-scale, bidirectional data and control network for EWHs. The first system (System A), focused on interoperability by following the SmartM2M standards. The second system (System B) focused on scalability and robustness by using tried-and-tested web technologies such as the MQTT protocol.

Secondly, a rigorous laboratory experiment was developed to autonomously perform precise EWH stimulation (water draw-off and heating), while making and collecting continuous measurement of temperatures, volumes and power at a one second period. The eight datasets generated using this setup were used for two distinct investigations: (1) to validate the nodal EWH models, thereby justifying them suitable for DSM application, and (2) to inspect the degree of energy savings made possible by constraining the heating of an EWH to a specific schedule.

7.1 Evaluation of findings

Based on the objectives stated in the Introduction (see page 6), the following paragraphs serves to evaluate the goals and outcomes of this study.

Objective 1: Develop a bidirectional communication system

In summary, System A consisted of 10 EWH hardware controllers and employed the OM2M implementation of the SmartM2M standards, using only the NSCL component and treating the EWH clients as legacy devices. System B consisted of 130 EWH hardware

controllers at the time of this writing and implemented the MQTT protocol using the Mosquitto MQTT broker and Paho MQTT client libraries.

1(a) Conform to Smart Grid and M2M/IoT vision: System A follows the REST architecture and is in line with SmartM2M. However, this is not the ideal design since an EWH should form part of a HAN, and use an M2M gateway to aggregate data. The cellular constraint inherently disregards this design. System B is not in line with the SmartM2M specifications, but proved to be a successful application. It was found unfeasible to use the current state of SmartM2M implementation for commercial application.

1(b) Use cellular access: Both System A and B used the hardware developed by Brown [25]. The cellular interface constraint proved challenging and required in-depth insight into the AT command philosophies for establishing internet connectivity. System A followed the *transparent* mode while System B followed the *opaque* mode. Although the opaque AT command mode required much more work than transparent mode it proved to be significantly more flexible, enabling the dialling of USSD codes, the MQTT protocol and performing over-the-air firmware updates.

1(c) Support both reporting and control: Both systems satisfied this requirement. System A was able to deliver data as a single UDP packet containing no binary data and inbound commands could only be received synchronously using a piggy-back mechanism. System B delivered and received data using a ongoing TCP connection which did not have restrictions on packet size or data format, and was able to receive inbound commands asynchronously.

Objective 2: Validate the Nodal models presented by Nel [26]

Both the one-node and two-node models where implemented in software. Simulations were performed using the collected experimental data as input in either an open-loop or closed loop simulation configuration. Model parameters where carefully determined using accurate measuring equipment and some additional lab experiments.

2(a) Design an experiment for model validation: A daily, periodic water draw-off routine and heating schedule were designed (Figure 5.2 on page 60). This was used consistently for all draw-off experiments. A periodic routine enabled calculating energy loss and efficiency over a 24 hour period of time without requiring knowledge of the edge conditions.

2(b) Implement a laboratory setup: A laboratory setup was designed and installed to execute the above water draw-off routine and heating schedule automatically and with precision (Figure 5.4 on page 65). The data collection and control infrastructure performed excellently and suffered no errors. However, a plumbing anomaly rendered one out of the eight main datasets partially unusable for analysis.

2(c) Determine the model parameters of the experimental EWH: All relevant EWH model parameters were measured with great care. This included, the element power, thermostat setpoint and hysteresis, and thermal resistance. The thermal resistance was measured using a static cooling test, and audited using a static heating test. Through the static tests, it was found that at maximum temperature ($\approx 65°C$), the EWHs emit approximately 93 Watt of power (2.23 kWh/day), which is roughly the same as a typical household incandescent light bulb.

2(d) Build a simulation platform: Both the one-node and two-node models where implemented using Python 3. All simulation were performed using Python 3 within *Jupyter notebook*. Experimental data was fed to simulations using a Python SQL API.

2(e) Evaluate thermal versus operational accuracy: The open-loop simulation was defined to validate the *thermal* behaviour of the EWH models. The closed-loop simulation was defined to validate the *operational* behaviour. No significant difference in performance could be observed between the two approaches, implying that inclusion of the modelled thermostat does not appear to effect the model behaviour.

2(f) Evaluate thermostat versus schedule control accuracy: Since the experimental methodology was designed in such a way to also be used for Objective 3, it meant that the model was validated using both thermostat and schedule control data. This coincidentally led to the unanticipated finding that the models behaved more predictably, and showed significantly better accuracy under schedule control.

2(g) Evaluate vertical versus horizontal orientation accuracy: No remarkable preference to EWH orientation could be detected. However, it may be noted at this point that the proposed experiments were not exhaustive to this end.

Objective 3: Determine the energy savings possible using basic scheduling

3(a) Design an experiment to measure energy efficiency: As stated earlier, the experiment from Objective 2 was also designed to allow evaluating the difference between thermostat and schedule control.

3(b) Must adhere to customer comfort constraint (above 49°C): A minimum heating time was determined to be 4 hours, and was sufficient to ensure the customer satisfaction requirement of 49°C was met (Table 5.1 on page 60). However, this proved to be conservative since the average water temperature under schedule control was 52.2°C.

3(c) Determine the conditions required to make a valid comparison: Although the customer satisfaction requirement was met, the average draw-off temperature for SC was 10°C lower than for TC. This meant that the efficiencies could not be directly compared, since the effective output was not the same. The comparison was performed using simulation and the validated models from Objective 2. Using the close-loop simulation, the simulated thermostat setpoint was iteratively set until an effective output temperature match was achieved. Using this method, it was found that the apparent 16% savings

observed during Objective 2 and other work from Literature, was a naive conclusion and was actually only an approximate of 6% savings.

3(d) Evaluate vertical versus horizontal orientation efficiency: Using the static heating test, a higher thermal resistance for the vertically mounted EWH was calculated (0.60°C/W on average), than for the horizontally mounted EWH (0.47°C/W on average). This resulted in a slightly higher efficiency (3.3% on average) observed for the vertically mounted EWH.

7.2 Recommendations

From the above conclusions, the following recommendations for future work can be made in terms of the M2M system design and the EWH experiment.

7.2.1 EWH M2M system implementation

Implement a home-area-network (HAN) and use a gateway aggregator: General consensus for both M2M and SG is that the architecture should consist of increasingly narrower sub-networks. Instead of using cellular, the EWH controller should be incorporated into a local private network using technologies such as Zigbee or PLC. This will also require implementing a gateway device to connect to the wide-area-network (WAN).

Investigate the use of CoAP: CoAP has been specifically designed to introduce the REST architecture to embedded environments and is inherently compatible with the oneM2M standards. MQTT on the other hand has not been. It may be valuable to investigate the case of CoAP.

Implement an IP stack on the microcontroller: Recent IoT embedded operating systems (such as Contiki [90]) ships with lightweight implementations of IPv4, IPv6, 6LoWPAN, UDP and TCP. By using these, modems may be used in dial-up mode and by doing so, eliminate the dependence on vendor specific AT command sets. This would greatly increase the flexibility and reusability of the software.

7.2.2 EWH experimentation:

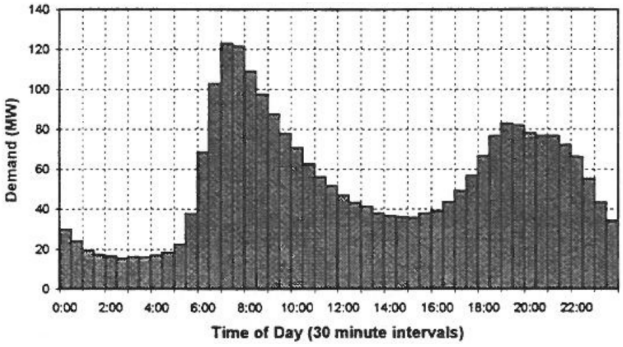
Investigate the effect different types of draw-off routines and heating schedules have on models: As stated earlier, this study only used one particular draw-off routine and heating schedule. Alternative usages also ought to be investigated to asses their effect on model behaviour and accuracy. For example, it may be valuable to investigate the difference between routines characterised by a few large events once a day, or many small events throughout the day. Additionally, further investigation may include finding a heating schedule that yields optimal draw-off energy for input energy given any draw-off routine.

Measure internal cylinder temperature: Most of the nodal model equations depend on knowledge of the internal cylinder temperature T_{cyl}. Being able to accurately

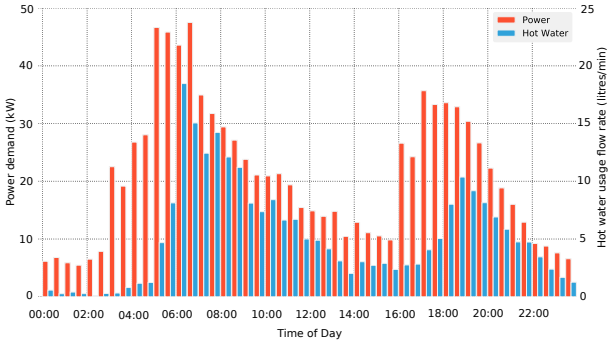
measure this temperature (as opposed to inferring from T_{out}) during experimentation might increase confidence in model behaviour.

Model EWH internal cylinder pressure: One of the datasets in this study has been rendered unusable due to heat being exchanged at the water inlet through the pressure relief valve. In order to account for this, a model of cylinder pressure is needed to calculate the additional energy loss.

Focus on peak-shaving and valley-filling: Figure 7.1b shows the average energy and water demand over 24 hours of approximately 100 EWHs using the data collected using system B [1]. Juxtaposing this to the notch test results from Eskom in 1997 in Figure 7.1a, the same double peak behaviour can be observed. However, since most of the users from System B have set a schedule to heat the water *before* usage, it has resulted in a 2 hour leading phase shift in energy demand. In other words, in an attempt to save energy by scheduling, users have moved the peak-load curve. It was shown in Objective 3 that for this particular experimental setup, only a 6% efficiency increase is possible using schedule control. Subsequently, it is recommended that bidirectional EWH control is rather used for peak-shaving and valley-filling schemes with customer comfort guarantee instead of standing-loss reduction schemes.



(a) Eskom notch test results in 1997 on approx. 120000 EWH



(b) System B results in 2016 on approx. 100 EWHs

Figure 7.1: Comparing daily EWH load profile 20 years after notch test program

[1] thanks to Marcel Roux (marcel.roux.email@gmail.com) for data extraction

Appendices

Appendix A

EWH controller hardware

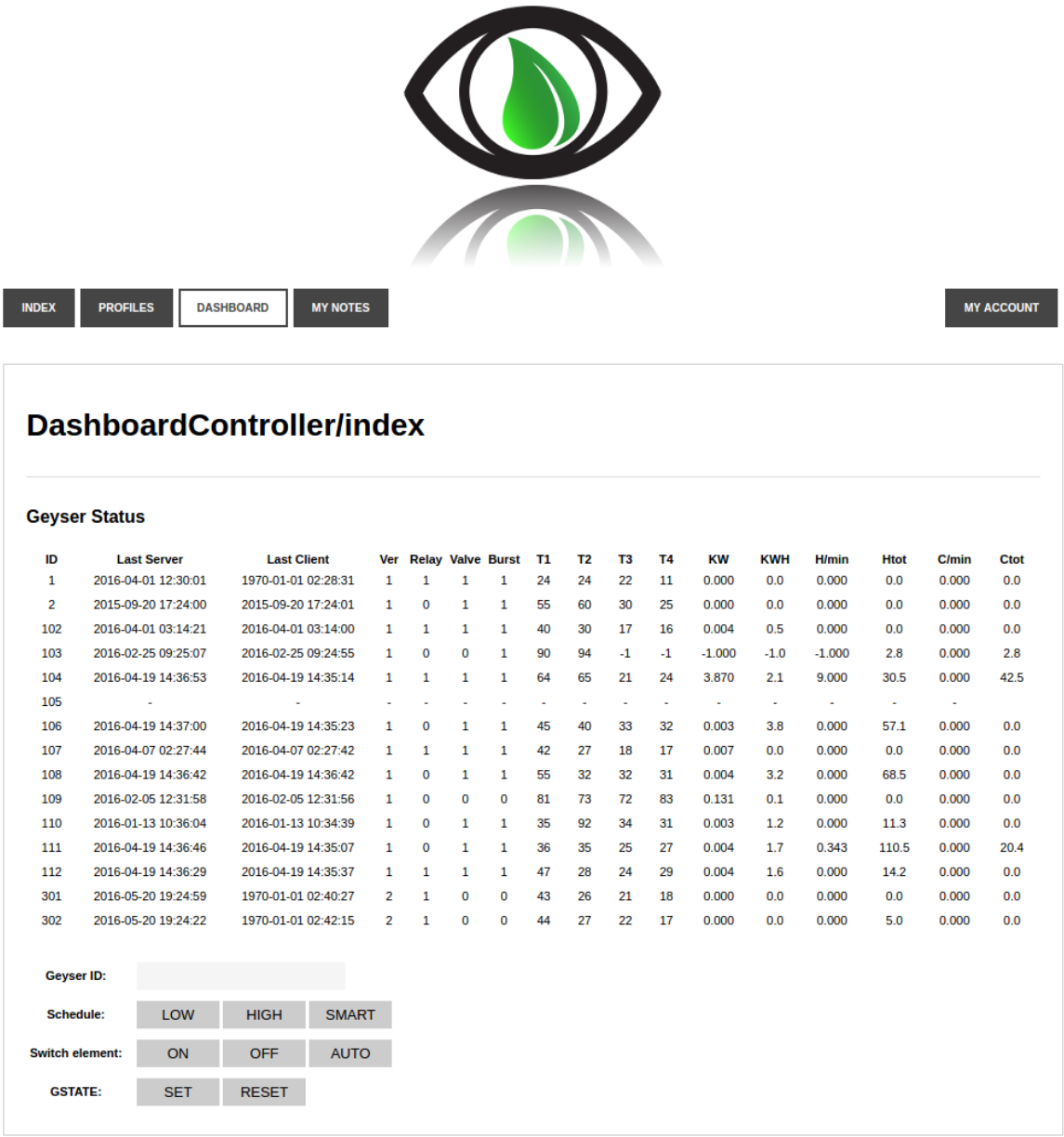
(a) Mk2 device with Maestro modem and Particle processor

(b) Mk3 EWH device with ublox modem and Atmel processor

Figure A.1: EWH monitoring and control devices

Appendix B

System A web interface



ID	Last Server	Last Client	Ver	Relay	Valve	Burst	T1	T2	T3	T4	KW	KWH	H/min	Htot	C/min	Ctot
1	2016-04-01 12:30:01	1970-01-01 02:28:31	1	1	1	1	24	24	22	11	0.000	0.0	0.000	0.0	0.000	0.0
2	2015-09-20 17:24:00	2015-09-20 17:24:01	1	0	1	1	55	60	30	25	0.000	0.0	0.000	0.0	0.000	0.0
102	2016-04-01 03:14:21	2016-04-01 03:14:00	1	1	1	1	40	30	17	16	0.004	0.5	0.000	0.0	0.000	0.0
103	2016-02-25 09:25:07	2016-02-25 09:24:55	1	0	0	1	90	94	-1	-1	-1.000	-1.0	-1.000	2.8	0.000	2.8
104	2016-04-19 14:36:53	2016-04-19 14:35:14	1	1	1	1	64	65	21	24	3.870	2.1	9.000	30.5	0.000	42.5
105	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
106	2016-04-19 14:37:00	2016-04-19 14:35:23	1	0	1	1	45	40	33	32	0.003	3.8	0.000	57.1	0.000	0.0
107	2016-04-07 02:27:44	2016-04-07 02:27:42	1	1	1	1	42	27	18	17	0.007	0.0	0.000	0.0	0.000	0.0
108	2016-04-19 14:36:42	2016-04-19 14:36:42	1	0	1	1	55	32	32	31	0.004	3.2	0.000	68.5	0.000	0.0
109	2016-02-05 12:31:58	2016-02-05 12:31:56	1	0	0	0	81	73	72	83	0.131	0.1	0.000	0.0	0.000	0.0
110	2016-01-13 10:36:04	2016-01-13 10:34:39	1	0	1	1	35	92	34	31	0.003	1.2	0.000	11.3	0.000	0.0
111	2016-04-19 14:36:46	2016-04-19 14:35:07	1	0	1	1	36	35	25	27	0.004	1.7	0.343	110.5	0.000	20.4
112	2016-04-19 14:36:29	2016-04-19 14:35:37	1	1	1	1	47	28	24	29	0.004	1.6	0.000	14.2	0.000	0.0
301	2016-05-20 19:24:59	1970-01-01 02:40:27	2	1	0	0	43	26	21	18	0.000	0.0	0.000	0.0	0.000	0.0
302	2016-05-20 19:24:22	1970-01-01 02:42:15	2	1	0	0	44	27	22	17	0.000	0.0	0.000	5.0	0.000	0.0

Figure B.1: System A web dashboard

Appendix C

System B web interface

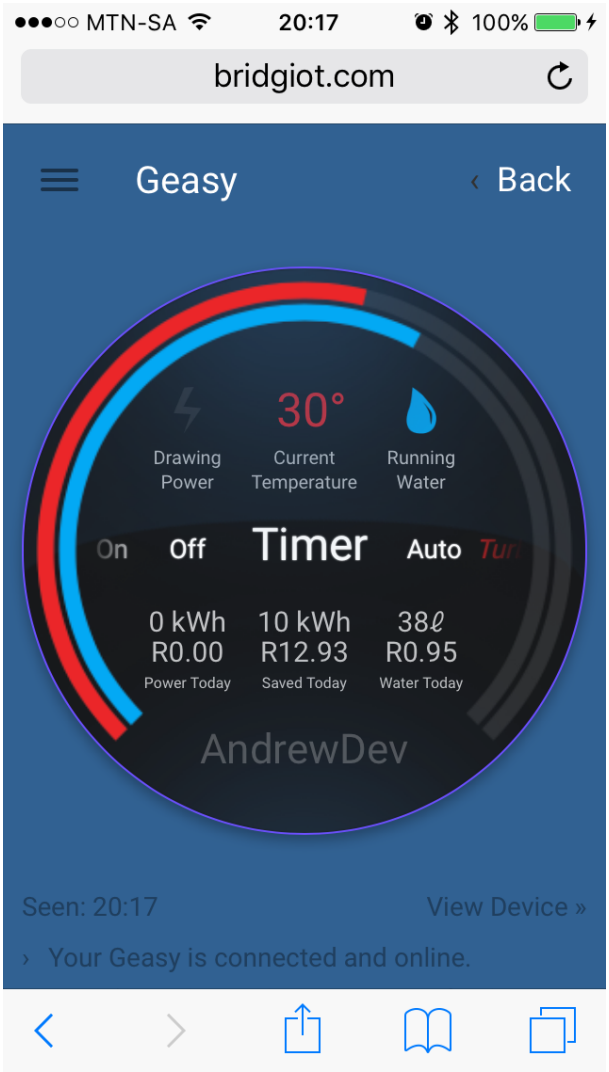
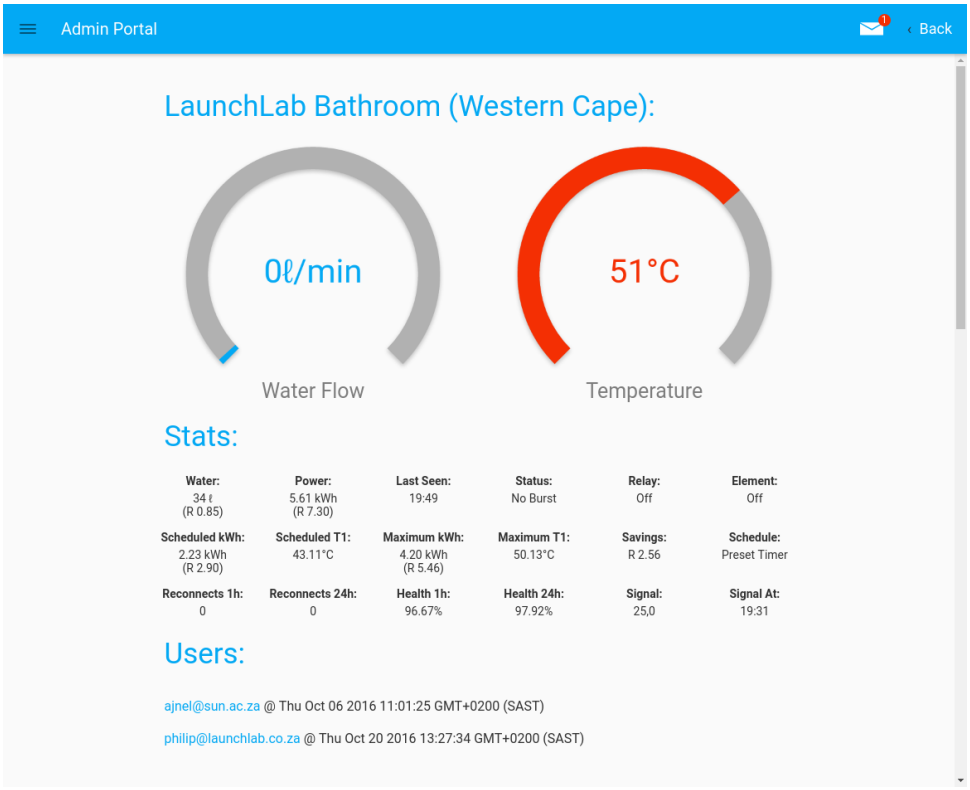
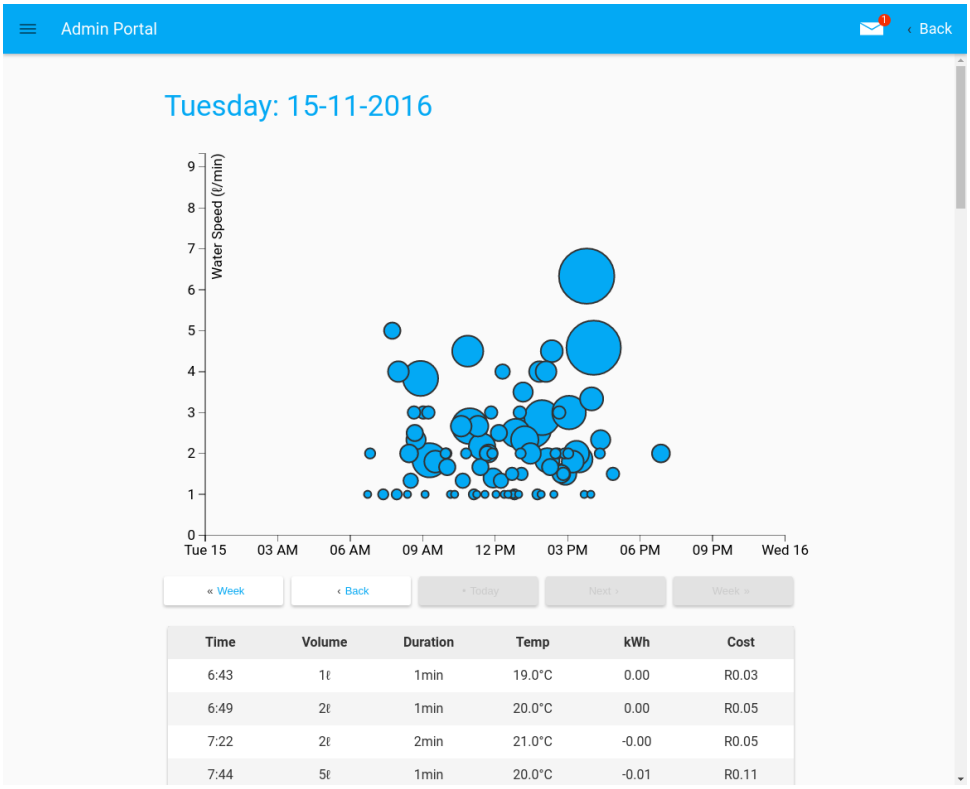


Figure C.1: System B client web interface optimised for mobile

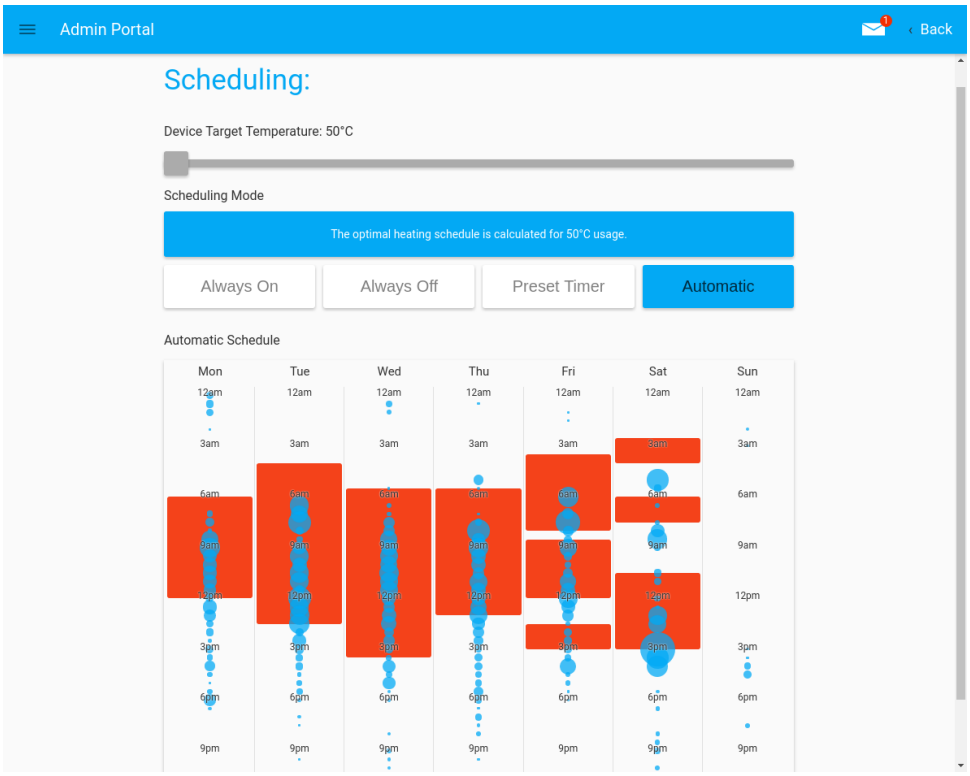


(a) Current state of EWH showing water draw-off flow rate and outlet temperature

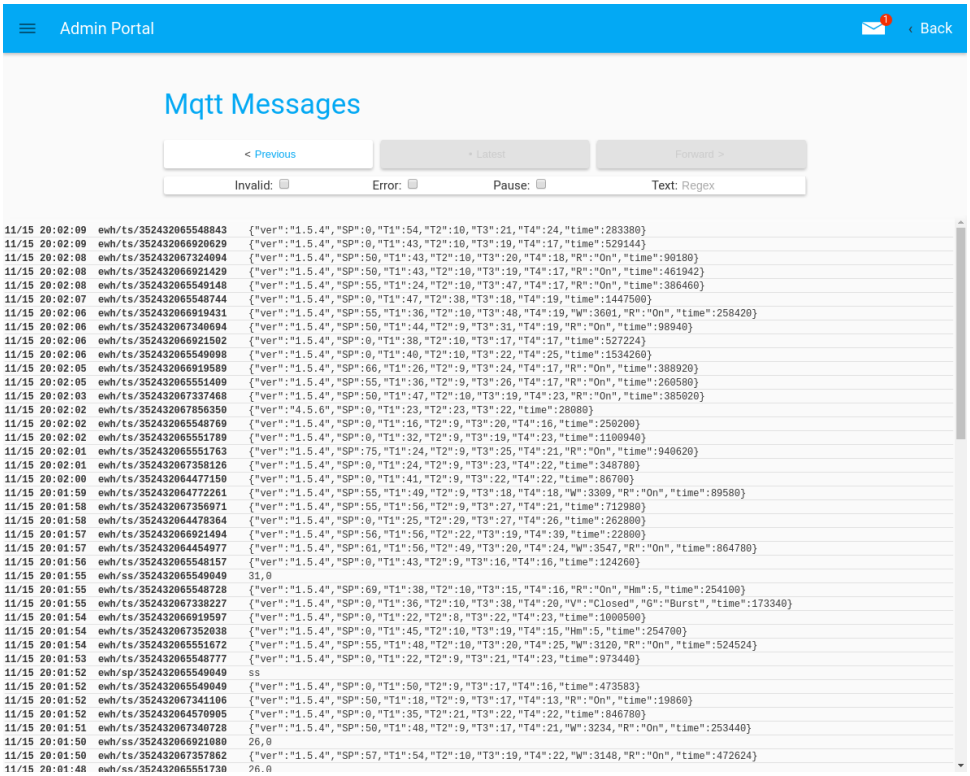


(b) Daily event breakdown

Figure C.2: System B admin web interface screen capture 1

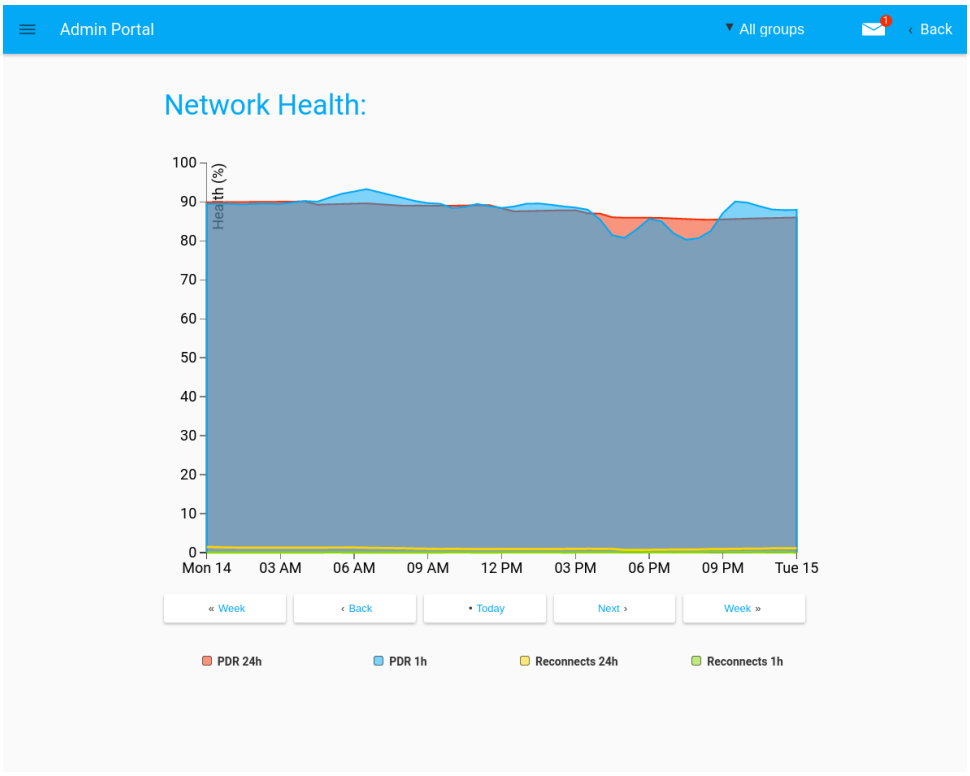


(a) Weekly heating schedule corresponding to water draw-off routine

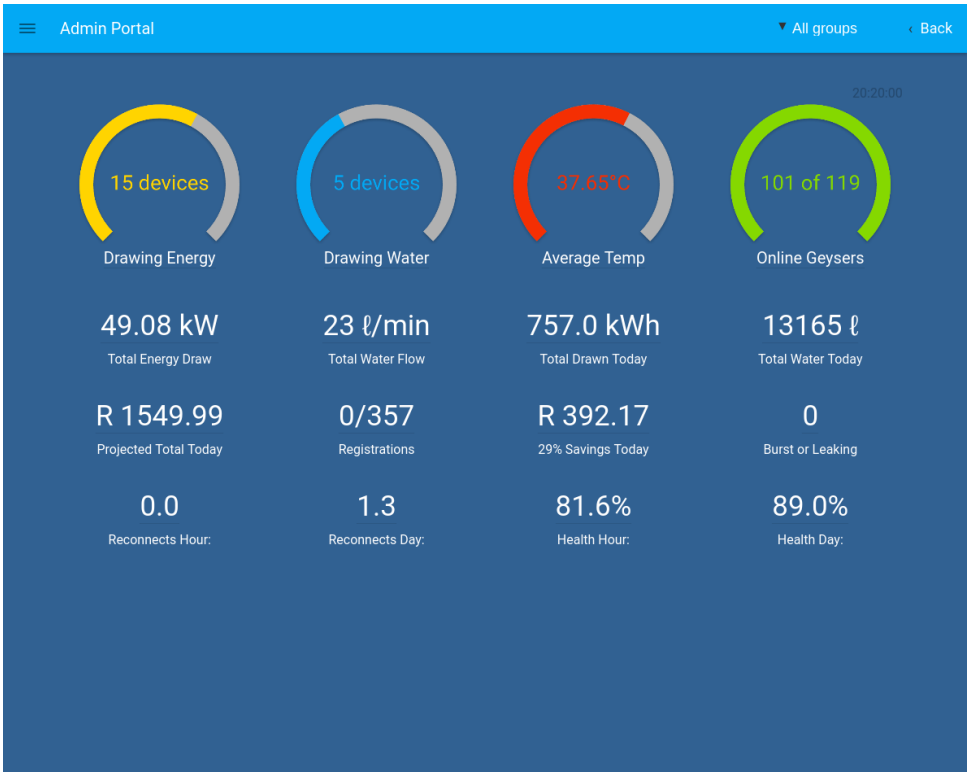


(b) Raw MQTT data from clients

Figure C.3: System B admin web interface screen capture 2



(a) Daily network performance



(b) Dashboard showing overview of complete system

Figure C.4: System B admin web interface screen capture 3

Appendix D

AT commands used

Module	ID	Discription	Command	Parameters
Net	AT		`AT`	(N/A)
	READ-COPS		`AT+COPS?`	(N/A)
	REG-COPS		`AT+COPS=0`	(N/A)
	DEREG-COPS		`AT+COPS=2`	(N/A)
	BALANCE		`ATD%s`	(N/A)
	NUMBER		`ATD%s`	(N/A)
	RSSI		`AT+CSQ`	(N/A)
	PLEASE-CM		`ATD%s%s#`	(N/A)
PDP	READ-PSD		`AT+UPSD=0,1`	(N/A)
	SET-PSD		`AT+UPSD=0,1,"internet"`	(N/A)
	ACT-PDP		`AT+UPSDA=0,3`	(N/A)
	DEAC-PDP		`AT+UPSDA=0,4`	(N/A)
Utils	READ-IMEI		`AT+CGSN`	(N/A)
	READ-CLOCK		`AT+CCLK?`	(N/A)
	RESOLVE-HOST		`AT+UDNSRN=0,"%s"`	(N/A)
TCP	TCP-SETUP		`AT+USOCR=6`	(N/A)
	TCP-CONNECT		`AT+USOCO=%d,"%s",%d`	Modem port; server IP addess; server-port
	TCP-PROMPT		`AT+USOWR=%d,%d`	Modem port, length of byte array
	TCP-SEND		`%s`	byte array
	TCP-READ		`AT+USORD=%d,%d`	Modem port, reply count
FTP	FTP-SETUP		`AT+UFTP=%d,"%s"`	action, parm
	FTP-SETUP-NUM		`AT+UFTP=%d,%d`	parm
	FTP-CTRL		`AT+UFTPC=%d`	action
	FTP-DWNL		`AT+UFTPC=4,"%s","%s"`	filename, filename
File	FILE-DELETE		`AT+UDELFILE="%s"`	filename

Table D.1: Summary of AT commands used in for the Mk3 communication stack driver. Refer to data sheet [116].

Appendix E

Modem driver

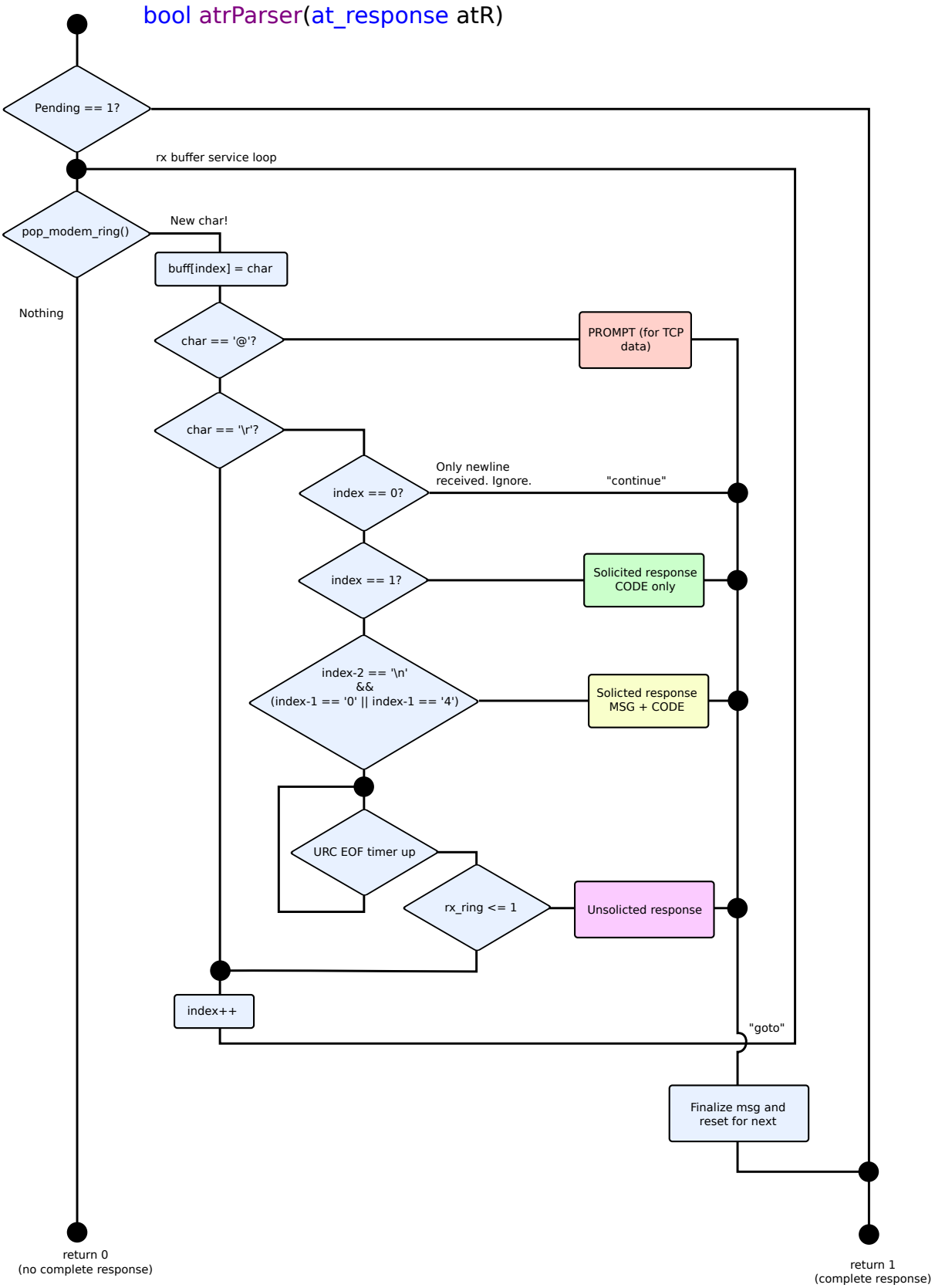


Figure E.1: AT response parser decision making tree

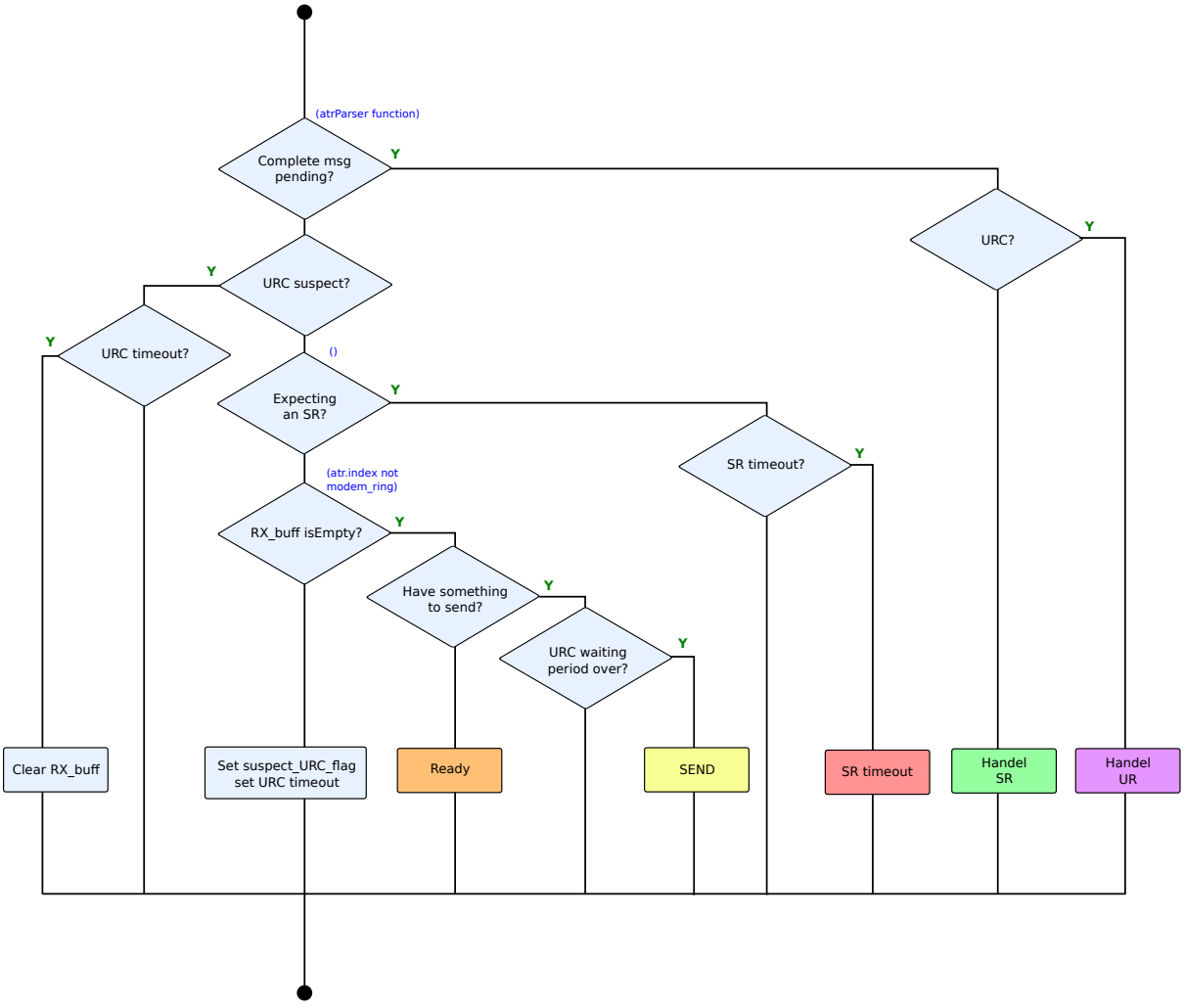


Figure E.2: Modem driver next action decision making tree. The *"next actions"* are shown in orange, yellow, red, green and purple. This decision making tree is particularly necessary since AT responses may occur both *synchronously* and *asynchronously*. (Note that the first decision is based on the result of the decision making tree from Figure E.1).

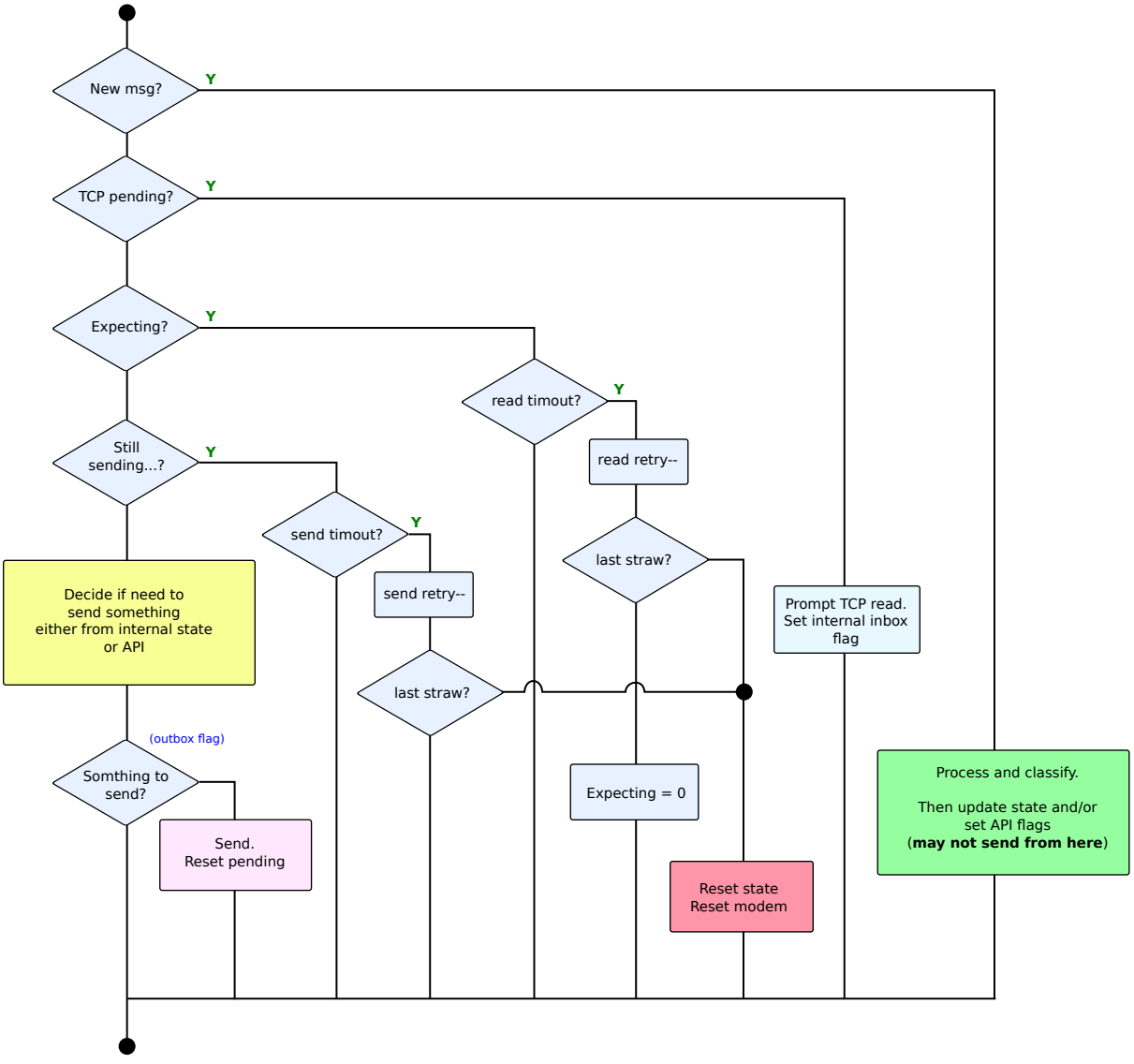


Figure E.3: MQTT service decision making tree

Appendix F

Raw EWH experiment plots

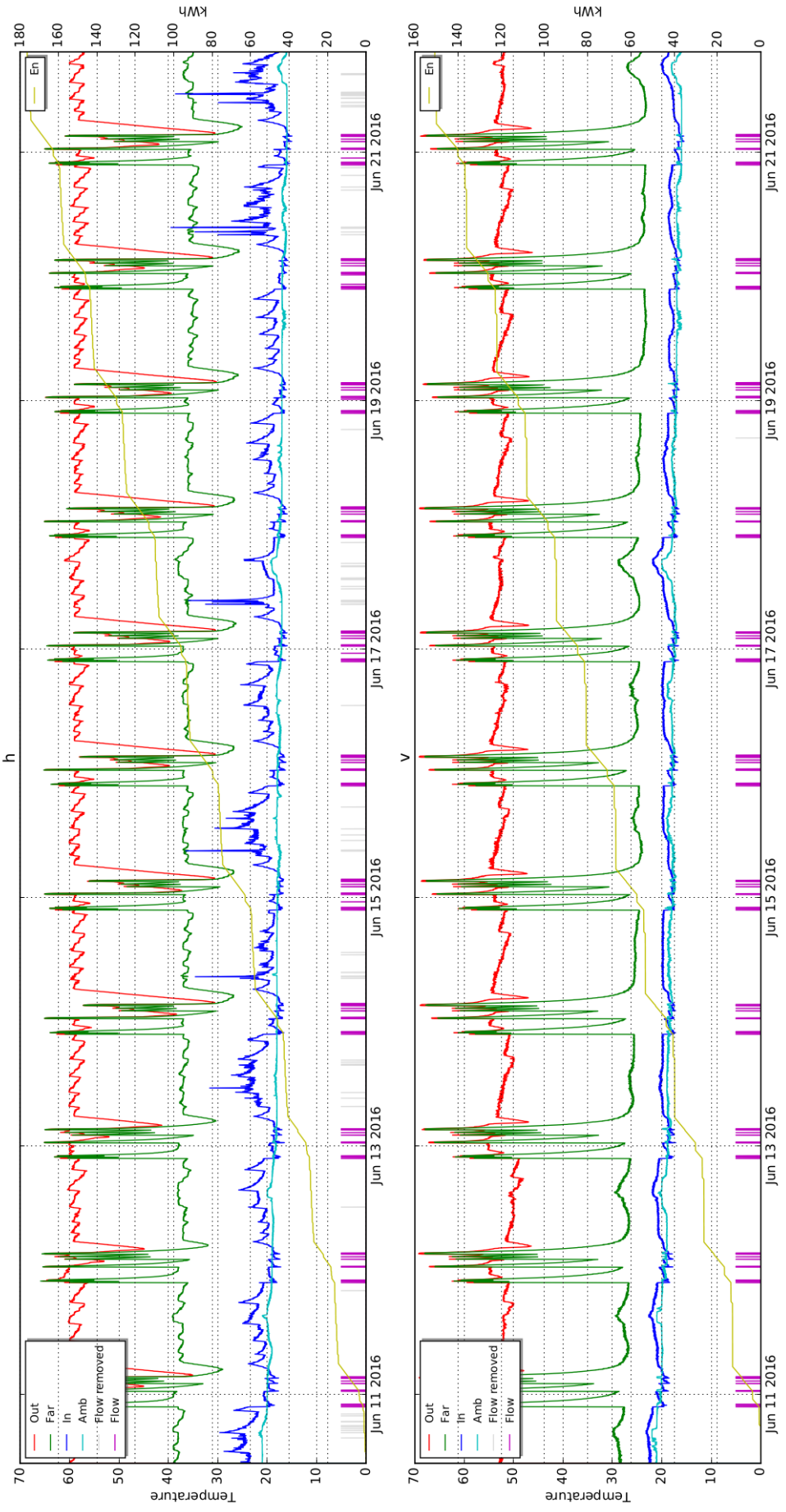


Figure F.1: Raw EWH experiment plot, Position A, Thermostat control

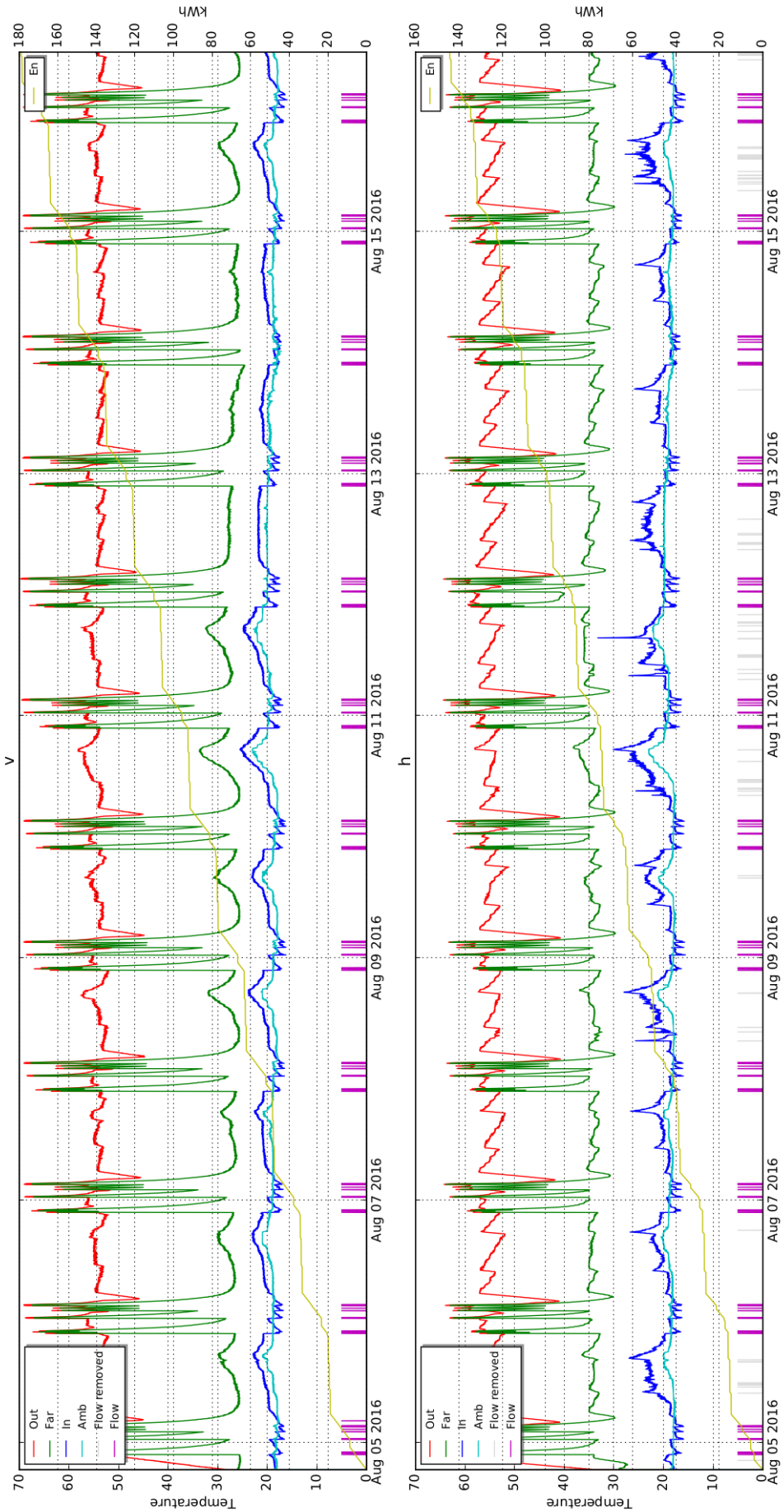


Figure F.2: Raw EWH experiment plot, Position B, Thermostat control

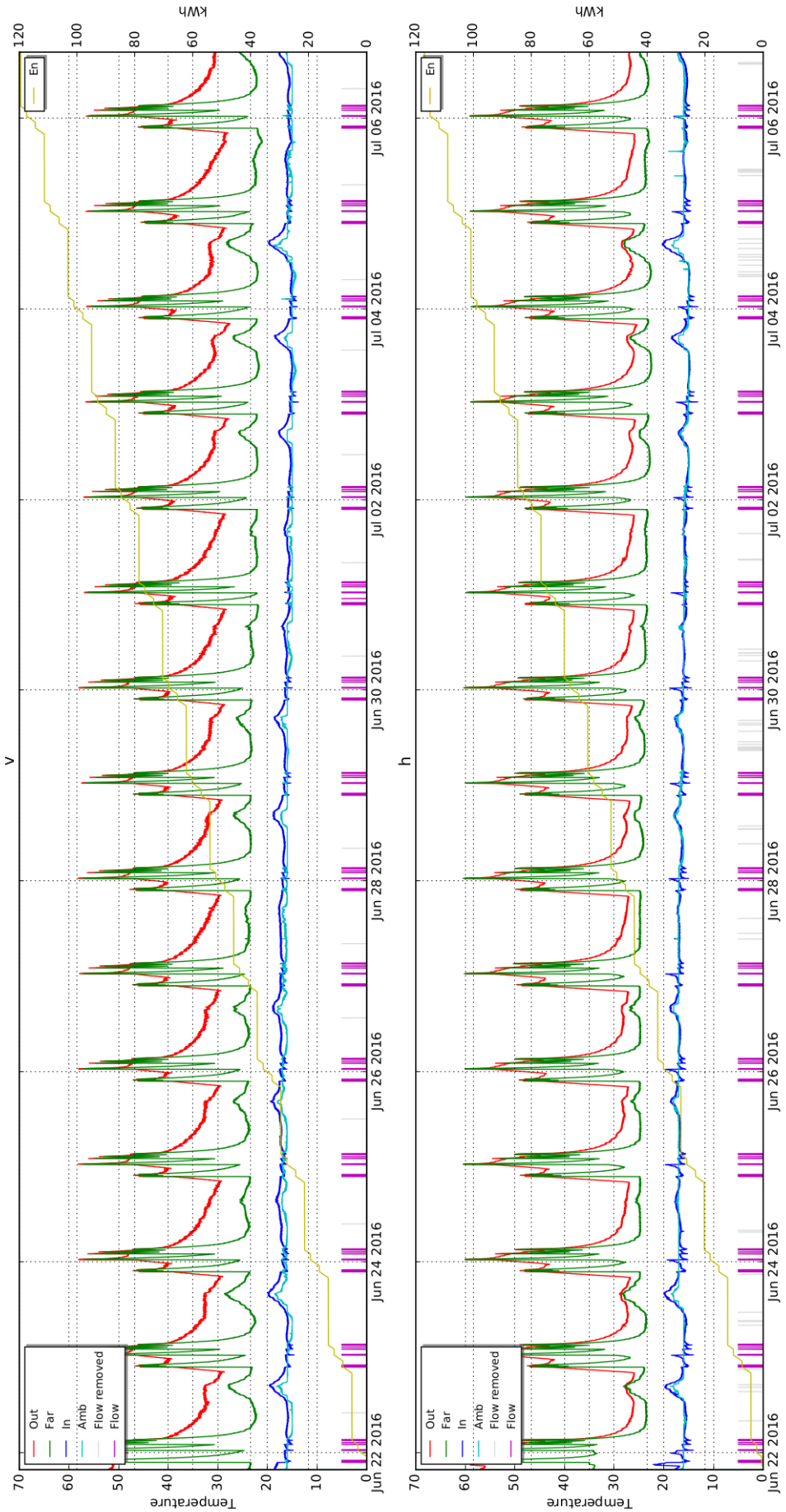


Figure F.3: Raw EWH experiment plot, Position A, Schedule control

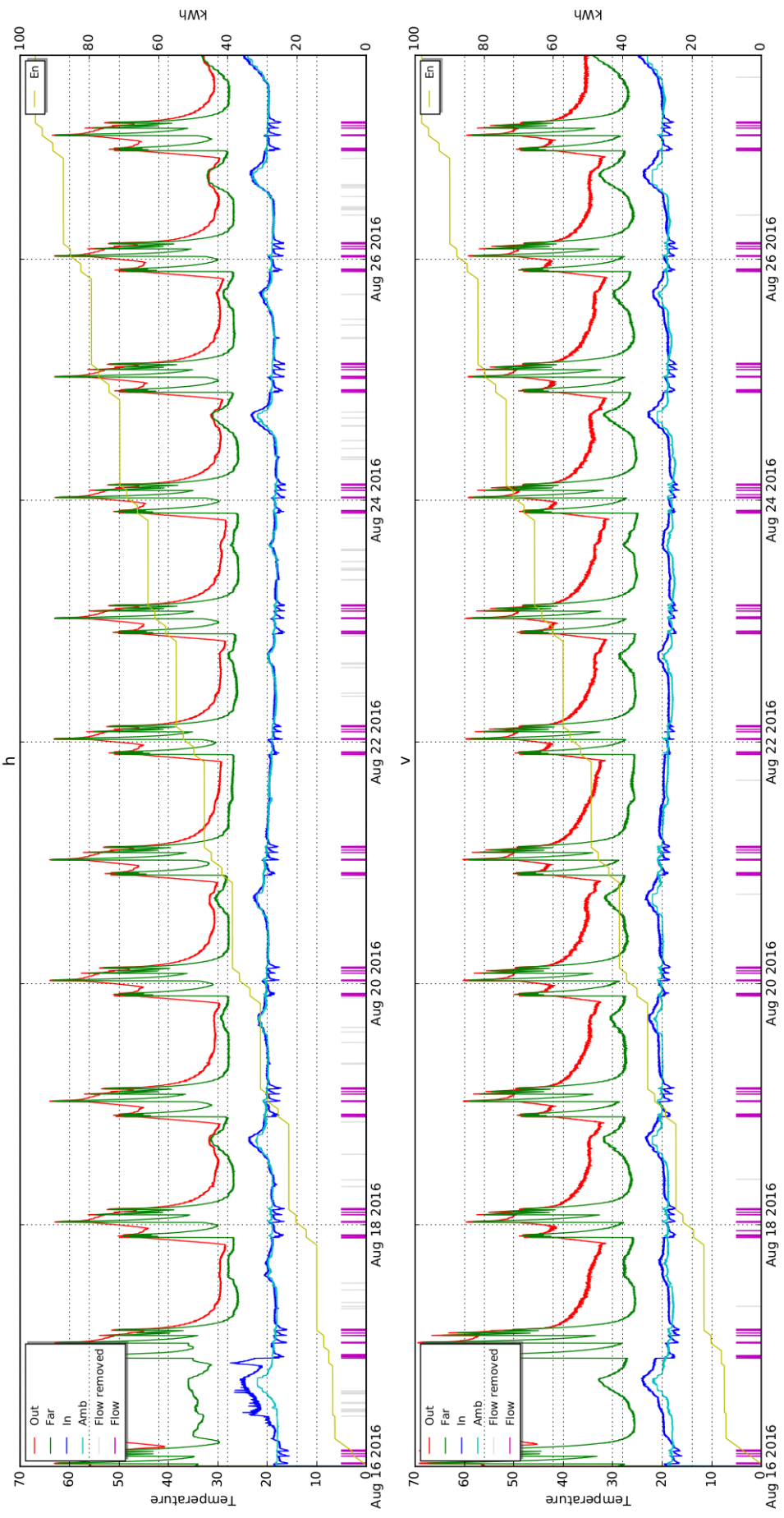


Figure F.4: Raw EWH experiment plot, Position B, Schedule control

Appendix G

Raw results for static cooldown and heating experiments

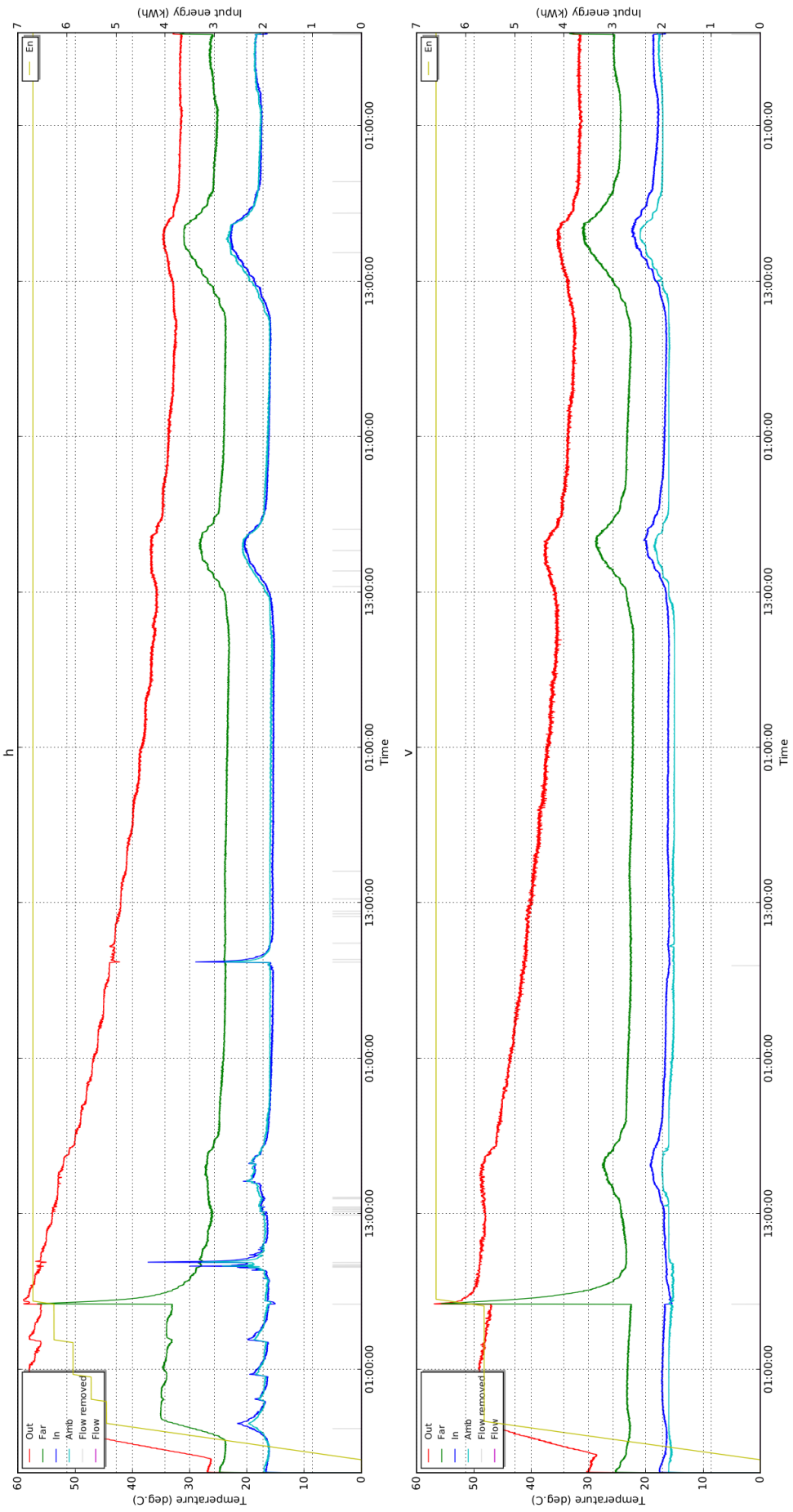


Figure G.1: Raw results, Position A, Static cooldown test

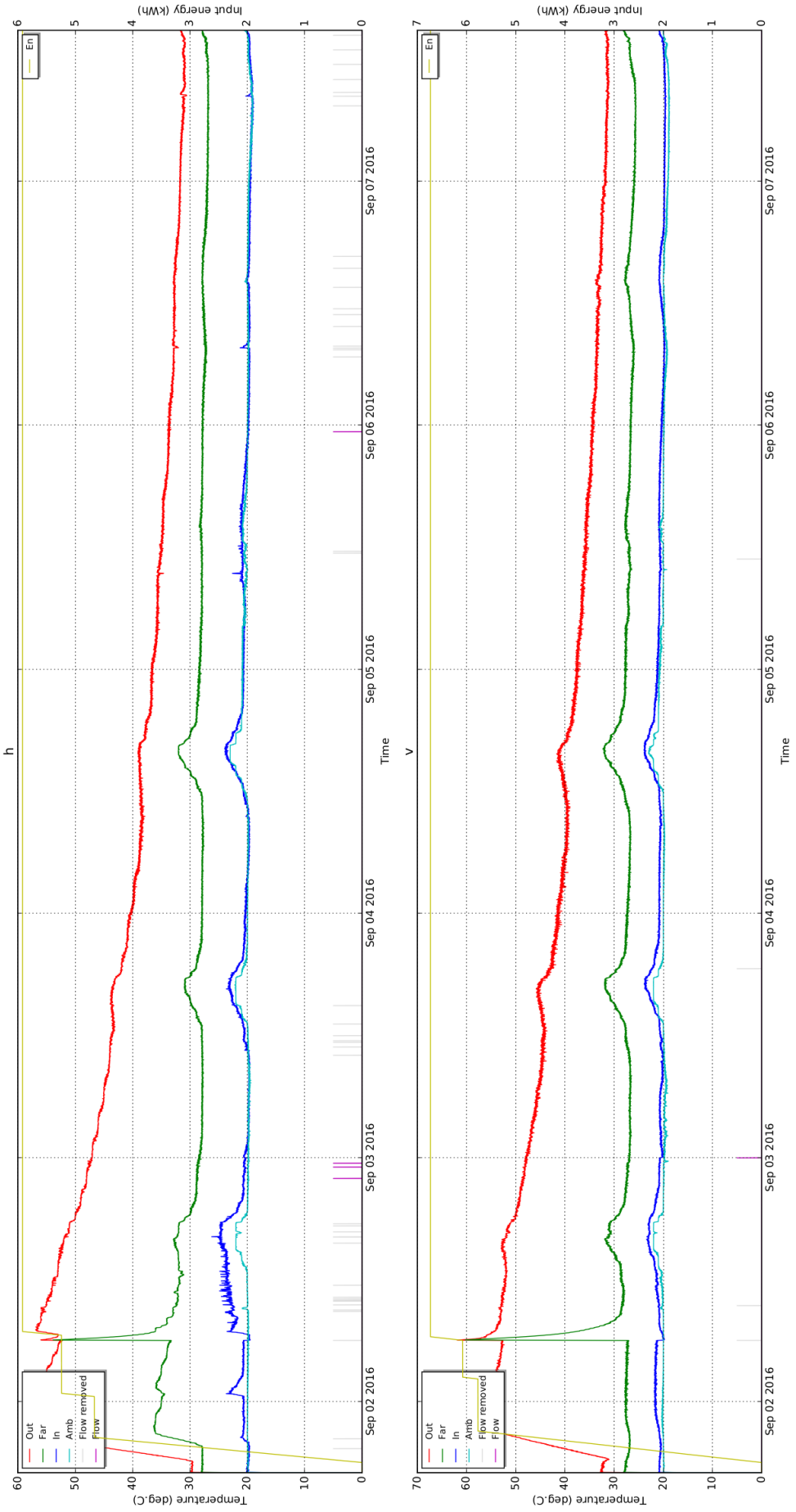


Figure G.2: Raw results, Position B, Static cooldown test

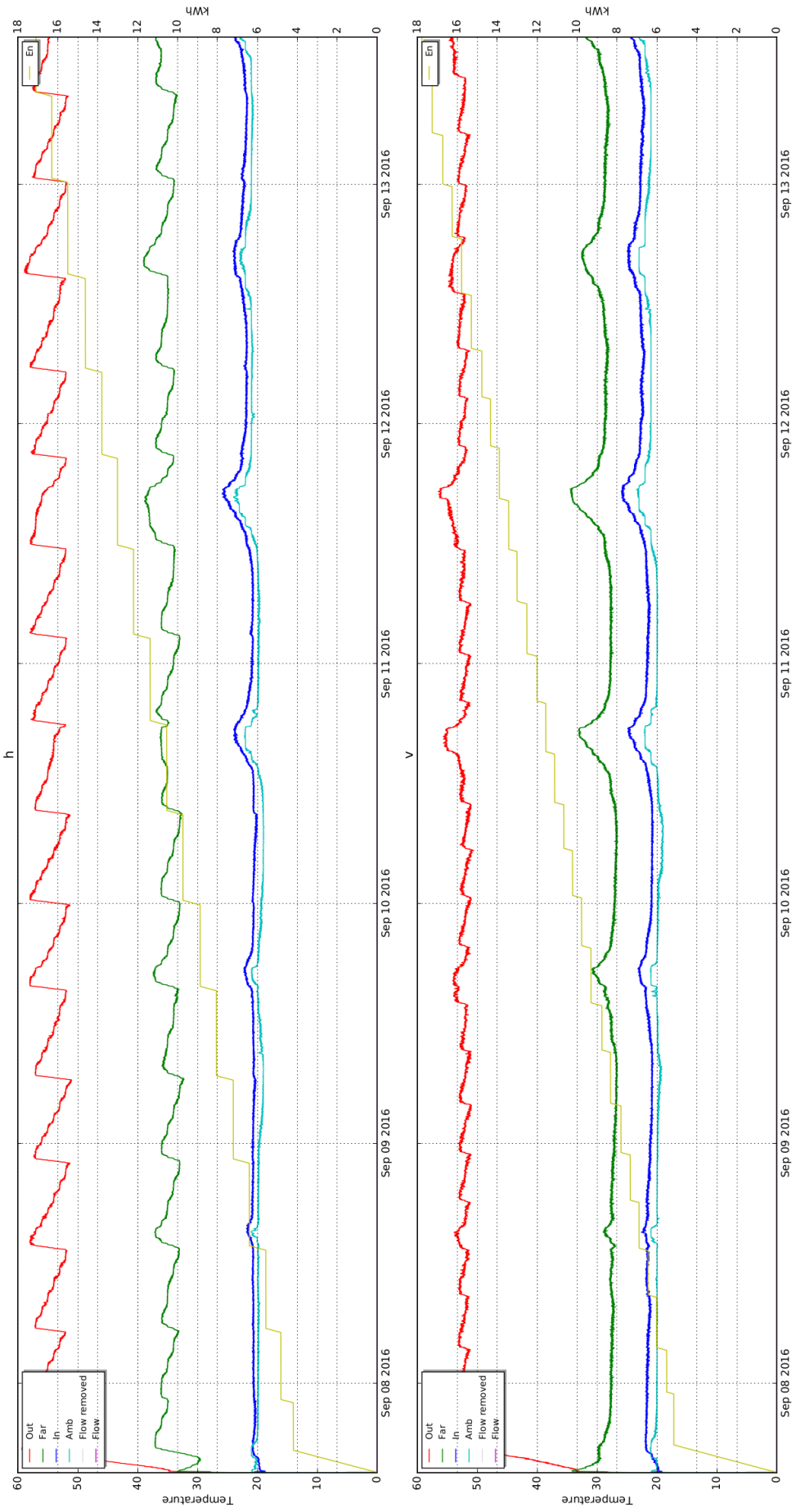


Figure G.3: Raw results, Position B, Static heating test

List of References

[1] "Global Energy Statistical Yearbook 2016," https://yearbook.enerdata.net/#electricity-domestic-consumption-data-by-region.html, accessed: Nov 2016.

[2] "BP Global Electricity - 2015 in review," http://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy/electricity.html, accessed: Nov 2016.

[3] N. M. Odhiambo, "Electricity consumption and economic growth in South Africa: A trivariate causality test," *Energy Economics*, vol. 31, no. 5, pp. 635–640, 2009.

[4] P. Nel, M. Booysen, and B. van der Merwe, "Energy perceptions in South Africa: An analysis of behaviour and understanding of electric water heaters," *Energy for Sustainable Development*, vol. 32, pp. 62–70, 2016.

[5] A. Sebitosi, "Energy efficiency, security of supply and the environment in south africa: Moving beyond the strategy documents," *Energy*, vol. 33, no. 11, pp. 1591–1596, 2008.

[6] "Department of Energy, South Africa Annual report," http://www.energy.gov.za/files/Annual%20Reports/DoE-Annual-Report-2015-16.pdf, accessed: Nov 2016.

[7] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Computer Networks*, vol. 56, no. 11, pp. 2742–2771, 2012.

[8] R. H. Khan and J. Y. Khan, "A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network," *Computer Networks*, vol. 57, no. 3, pp. 825–845, 2013.

[9] H. T. Haider, O. H. See, and W. Elmenreich, "A review of residential demand response of smart grid," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 166–178, 2016.

[10] NIST, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0," https://www.nist.gov/sites/default/files/documents/smartgrid/NIST_Framework_Release_2-0_corr.pdf, accessed: October 2016.

[11] M. C. Falvo, L. Martirano, D. Sbordone, and E. Bocci, "Technologies for smart grids: a brief review," in *Environment and Electrical Engineering (EEEIC), 2013 12th International Conference on*. IEEE, 2013, pp. 369–375.

[12] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid," *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, 2011.

[13] E. Ancillotti, R. Bruno, and M. Conti, "The role of communication systems in smart grids: Architectures, technical solutions and research challenges," *Computer Communications*, vol. 36, no. 17, pp. 1665–1697, 2013.

[14] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, apr 2014. [Online]. Available: http://dx.doi.org/10.1007/s10796-014-9492-7

[15] "How the Next Evolution of the Internet Is Changing Everything," http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, accessed: October 2016.

[16] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 53–59, 2011.

[17] ETSI, "Machine-to-Machine communications (M2M); Applicability of M2M architecture to Smart Grid Networks TR 102 935 V2.1.1," European Telecommunications Standards Institute, Technical Report, Sept 2012.

[18] "Eskom Geyser fact sheet 2013," http://www.eskom.co.za/sites/IDM/Documents/1317_geyser_fact_sheet_no_rmr.pdf, accessed: Nov 2016.

[19] N. Beute and G. Delport, "An historic overview of controlling domestic water heating," in *Proceedings of the 14th Domestic Use of Energy Conference*. Citeseer, 2006, pp. 41–46.

[20] "Intensive energy users group - Tariff and Subsidy Infographics," http://www.eiug.org.za/infographics/, accessed: Nov 2016.

[21] "Schedule of standard prices from Eskom tariffs 2015," http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/Eskom%20Schedule%20of%20Std%20Prices%202015_16%20(29%20Jan%202015).pdf, accessed: Nov 2016.

[22] H. Naude, "Demand-side management by centralized ripple control system," in *Proceedings of the 2nd Domestic Use of Energy Conference*. Citeseer, 1995, pp. 71–82.

[23] K. Kostková, L. Omelina, P. Kyčina, and P. Jamrich, "An introduction to load management," *Electric Power Systems Research*, vol. 95, pp. 184–191, 2013.

[24] C. Forlee, "Water heating in South Africa facts and figures from the 1997" notch testing" program," in *5th International Domestic Use of Electrical Energy Conference, Cape Town*, 1998, pp. 265–270.

[25] J.W.K Brown, "Design and implementation of an intelligent water heater control module for feedback and demnad-side management," Master's thesis, University of Stellenbosch, November 2015.

[26] P.J.C Nel, "Rethinking electrical water heaters," Master's thesis, University of Stellenbosch, September 2015.

[27] M.J. Booysen and J.A.A. Engelbrecht and A. Molinaro, "Proof of concept : large-scale monitor and control of household water heating in near real-time," *International Conference on Applied Energy ICAE 2013, Jul 1-4, 2013, Pretoria, South Africa*, 2013.

[28] B. Lévesque, M. Lavoie, and J. Joly, "Residential water heater temperature: 49 or 60 degrees celsius?" *Canadian Journal of Infectious Diseases*, vol. 15, no. 1, pp. 11–12, 2004.

[29] "South African Water Research Commission," http://www.wrc.org.za/, accessed: May 2015.

[30] "Mkhondo Local Municipality," http://www.mkhondo.gov.za/, accessed: October 2016.

[31] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE transactions on industrial informatics*, vol. 7, no. 3, pp. 381–388, 2011.

[32] L. C. Siebert, L. R. Ferreira, E. K. Yamakawa, E. S. Custodio, A. R. Aoki, T. S. Fernandes, and K. H. Cardoso, "Centralized and decentralized approaches to demand response using smart plugs," in *T&D Conference and Exposition, 2014 IEEE PES*. IEEE, 2014, pp. 1–5.

[33] S. Lu, N. Samaan, R. Diao, M. Elizondo, C. Jin, E. Mayhorn, Y. Zhang, and H. Kirkham, "Centralized and decentralized control for demand response," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*. IEEE, 2011, pp. 1–8.

[34] P. Siano, "Demand response and smart grids: A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, 2014.

[35] "IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads," IEEE, IEEE Std 2030-2011, Sep 2011.

[36] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, 2011.

[37] M. Kuzlu and M. Pipattanasomporn, "Assessment of communication technologies and network requirements for different smart grid applications," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*. IEEE, 2013, pp. 1–6.

[38] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Analysis of communication schemes for Advanced Metering Infrastructure (AMI)," in *2014 IEEE PES General Meeting| Conference & Exposition*. IEEE, 2014, pp. 1–5.

[39] K. Ma, T. Yao, J. Yang, and X. Guan, "Residential power scheduling for demand response in smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 320–325, 2016.

[40] C. Pirak, T. Sangsuwan, and S. Buayairaksa, "Recent advances in communication technologies for smart grid application: A review," in *Electrical Engineering Congress (iEECON), 2014 International*. IEEE, 2014, pp. 1–4.

[41] A. Usman and S. H. Shami, "Evolution of communication technologies for smart grid applications," *Renewable and Sustainable Energy Reviews*, vol. 19, pp. 191–199, 2013.

[42] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. P. Chen, "A survey of communication/networking in Smart Grids," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 391–404, 2012.

[43] M. G. Kanabar, I. Voloh, and D. McGinn, "A review of smart grid standards for protection, control, and monitoring applications," in *Protective Relay Engineers, 2012 65th Annual Conference for*. IEEE, 2012, pp. 281–289.

[44] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart grid technologies: communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[45] F. F. Wu, K. Moslehi, and A. Bose, "Power system control centers: Past, present, and future," *Proceedings of the IEEE*, vol. 93, no. 11, pp. 1890–1908, 2005.

[46] ETSI, "Machine-to-Machine communications (M2M); M2M service requirements TS 102 689 V2.1.1," European Telecommunications Standards Institute, Technical Specifications, Jul 2013.

[47] ETSI, "Machine-to-Machine communications (M2M): Functional architecture TS 102 690 V2.1.1," European Telecommunications Standards Institute, Technical Specifications, Oct 2013.

[48] ETSI, "Machine-to-Machine communications (M2M);mIa, dIa and mId interfaces TS 102 921 V1.3.1," European Telecommunications Standards Institute, Technical Specifications, Sept 2014.

[49] H. Anton-Haro and M. Dohler, *Machine-to-machine (M2M) communications : architecture, performance and applications*. Cambridge, UK Waltham, MA: Woodhead Publishing, 2015.

[50] M. B. Alaya, Y. Banouar, T. Monteil, C. Chassot, and K. Drira, "OM2M: Extensible ETSI-compliant M2M service platform with self-configuration capability," *Procedia Computer Science*, vol. 32, pp. 1079–1086, 2014.

[51] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common m2m service layer platform: Introduction to onem2m," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[52] oneM2M, "onem2m white paper - the interoperability enabler for the entire m2m and iot ecosystem," oneM2M, Tech. Rep., 2015.

[53] "OneM2M," http://www.onem2m.org/, accessed: Nov 2015.

[54] J. Song, A. Kunz, M. Schmidt, and P. Szczytowski, "Connecting and managing m2m devices in the future internet," *Mobile Networks and Applications*, vol. 19, no. 1, pp. 4–17, 2014.

[55] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M communications a systems approach*. Hoboken, N.J: Wiley, 2012.

[56] ETSI, "Machine-to-Machine communications (M2M); Smart Metering Use Cases TR 102 691 V1.1.1," European Telecommunications Standards Institute, Technical Report, May 2010.

[57] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[58] "OM2M - Connecting things," www.eclipse.org/om2m/, accessed: Feb 2015.

[59] M. Meddeb, M. B. Alaya, T. Monteil, A. Dhraief, and K. Drira, "M2M platform with autonomic device management service," *Procedia Computer Science*, vol. 32, pp. 1063–1070, 2014.

[60] A. Elmangoush, H. Coskun, S. Wahle, and T. Magedanz, "Design aspects for a reference M2M communication platform for Smart Cities," in *Innovations in Information Technology (IIT), 2013 9th International Conference on*. IEEE, 2013, pp. 204–209.

[61] A. Elmangoush, R. Steinke, A. Al-Hezmi, and T. Magedanz, "On the usage of standardised m2m platforms for smart energy management," in *The International Conference on Information Networking 2014 (ICOIN2014)*. IEEE, 2014, pp. 79–84.

[62] A. Elmangoush, R. Steinke, T. Magedanz, A. A. Corici, A. Bourreau, and A. Al-Hezmi, "Application-derived communication protocol selection in M2M platforms for smart cities," *2015 18th International Conference on Intelligence in Next Generation Networks*, pp. 76–82, 2015.

[63] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, "Toward semantic interoperability in onem2m architecture," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 35–41, 2015.

[64] "IPSO Alliance," http://www.ipso-alliance.org/, accessed: October 2016.

[65] "OMA LWM2M," http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0, accessed: June 2016.

[66] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. [Online]. Available: http://dx.doi.org/10.1109/COMST.2015.2444095

[67] A. Talaminos-Barroso, M. A. Estudillo-Valderrama, L. M. Roa, J. Reina-Tosina, and F. Ortega-Ruiz, "A Machine-to-Machine protocol benchmark for eHealth applications - Use case: Respiratory rehabilitation," *Computer Methods and Programs in Biomedicine*, vol. 129, pp. 1–11, jun 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cmpb.2016.03.004

[68] OASIS, "Advanced Message Queuing Protocol (AMQP) Version 1.0," OASIS, http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf, Normative standard, Oct 2012.

[69] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF, Tech. Rep. 7252, June 2014.

[70] O. M. Group, "Data Distribution Service (DDS) 1.4," OMG, Normative standard, April 2015. [Online]. Available: http://www.omg.org/spec/DDS/1.4/

[71] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing," Internet Requests for Comments, RFC Editor, RFC 7230, June 2014, http://www.rfc-editor.org/rfc/rfc7230.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7230.txt

[72] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," Internet Requests for Comments, RFC Editor, RFC 7231, June 2014, http://www.rfc-editor.org/rfc/rfc7231.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7231.txt

[73] B. Cohn, Coppen and Gupta, "MQTT Version 3.1.1," OASIS, Normative standard, Oct 2014.

[74] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Address Format," Internet Requests for Comments, RFC Editor, RFC 7622, September 2015.

[75] "MQTT," http://mqtt.org/, accessed: May 2016.

[76] Stanford-Clark and L. Truong, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2," OASIS, Normative standard, Nov 2013.

[77] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-SN: A publish/subscribe protocol for Wireless Sensor Networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on.* IEEE, 2008, pp. 791–798.

[78] "Building Facebook Messenger," https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920, accessed: Oct 2016.

[79] R. V. Chander, S. Elias, S. Shivashankar, and P. Manoj, "A REST based design for Web of Things in smart environments," in *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on.* IEEE, 2012, pp. 337–342.

[80] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Internet Requests for Comments, RFC Editor, RFC 4919, August 2007, http://www.rfc-editor.org/rfc/rfc4919.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4919.txt

[81] "Open Mobile Alliance Device Management," http://openmobilealliance.org/about-oma/work-program/device-management/, accessed: June 2015.

[82] C. Pereira and A. Aguiar, "Towards Efficient Mobile M2M Communications: Survey and Open Challenges," *Sensors*, vol. 14, no. 10, pp. 19 582–19 608, oct 2014. [Online]. Available: http://dx.doi.org/10.3390/s141019582

[83] E. Davis, A. Calveras, and I. Demirkol, "Improving Packet Delivery Performance of Publish/Subscribe Protocols in Wireless Sensor Networks," *Sensors*, vol. 13, no. 1, pp. 648–680, jan 2013. [Online]. Available: http://dx.doi.org/10.3390/s130100648

[84] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on.* IEEE, 2014, pp. 1–6.

[85] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," in *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on.* IEEE, 2013, pp. 1–6.

[86] "CoAP Online," http://coap.technology/, accessed: June 2015.

[87] "Paho," http://www.eclipse.org/paho/, accessed: May 2016.

[88] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," Internet Requests for Comments, RFC Editor, RFC 4944, September 2007, http://www.rfc-editor.org/rfc/rfc4944.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4944.txt

[89] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37–45, 2008.

[90] "Contiki OS," http://www.contiki-os.org/, accessed: May 2015.

[91] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.

[92] A. Dunkels, "Full TCP/IP for 8-bit architectures," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 85–98.

[93] A. Dunkels, "Design and Implementation of the lwIP TCP/IP Stack," *Swedish Institute of Computer Science*, vol. 2, p. 77, 2001.

[94] P. Du and N. Lu, "Appliance commitment for household load scheduling," *Smart Grid, IEEE Transactions on*, vol. 2, no. 2, pp. 411–419, 2011.

[95] C. Diduch, M. Shaad, R. Errouissi, M. Kaye, J. Meng, and L. Chang, "Aggregated domestic electric water heater control - building on smart grid infrastructure," *2012 IEEE 7th International Power Electronics and Motion Control Conference - ECCE Asia conference proceedings : June 2-5, 2012, Harbin, China*, 2012.

[96] M. Shaad, A. Momeni, C. P. Diduch, M. Kaye, and L. Chang, "Parameter identification of thermal models for domestic electric water heaters in a direct load control program," in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012, pp. 1–5.

[97] A. Afram and F. Janabi-Sharifi, "Review of modeling methods for HVAC systems," *Applied Thermal Engineering*, vol. 67, no. 1, pp. 507–519, 2014.

[98] H. Njoku, O. Ekechukwu, and S. Onyegegbu, "Analysis of stratified thermal storage systems: An overview," *Heat and Mass Transfer*, vol. 50, no. 7, pp. 1017–1030, 2014.

[99] M. H. Nehrir, R. Jia, D. A. Pierre, and D. J. Hammerstrom, "Power management of aggregate electric water heater loads by voltage control," in *2007 IEEE Power Engineering Society General Meeting*, 2007.

[100] R. Diao, S. Lu, M. Elizondo, E. Mayhorn, Y. Zhang, and N. Samaan, "Electric water heater modeling and control strategies for demand response," in *2012 IEEE Power and Energy Society General Meeting*. IEEE, 2012, pp. 1–8.

[101] Y. Cengel and A. Ghajar, *Heat and Mass Transfer: Fundamentals and Applications*. McGraw-Hill, 2010.

[102] P.J.C. Nel, M.J. Booysen, A.B. van der Merwe, "A computationally inexpensive energy model for horizontal electric water heaters with scheduling," in *IEEE Transactions on Smartgrid*, 2016.

[103] "Particle," https://www.particle.io/, accessed: March 2015.

[104] "AtmelXmega128A4U," http://www.atmel.com/devices/ATXMEGA128A4U.aspx, accessed: March 2016.

[105] "The Modem of Dennis Hayes and Dale Heatherington," http://history-computer.com/ModernComputer/Basis/modem.html, accessed: June 2015.

[106] Y. Rekhter and T. Li, "The Point-to-Point Protocol (PPP)," Internet Requests for Comments, RFC Editor, RFC 1661, July 1994. [Online]. Available: https://tools.ietf.org/html/rfc1661

[107] "OM2M Java Client API - Github," https://github.com/BeliliFahem/om2m-java-client-api, accessed: April 2015.

[108] "Apache HTTP server project," https://httpd.apache.org/, accessed: April 2015.

[109] "Mosquitto Online," https://mosquitto.org/, accessed: June 2016.

[110] "Node JS," https://nodejs.org/en/, accessed: Nov 2016.

[111] "MeteorJS," https://www.meteor.com/, accessed: Nov 2016.

[112] u blox, "AT Commands Examples: Application Note," u-blox, Tech. Rep., Jul 2015. [Online]. Available: https://www.u-blox.com/sites/default/files/AT-CommandsExamples_AppNote_(UBX-13001820).pdf

[113] M.J. Booysen and A.H. Cloete, "Sustainability through Intelligent Scheduling of Electric Water Heaters in a Smart Grid," in *Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/-CyberSciTech), 2016 IEEE 14th Intl C.* IEEE, 2016, pp. 848–855.

[114] "The installation, maintenance, replacement and repair of fixed electric storage water heating systems," SANS, SANS 10254, February 2012.

[115] "Kwikot," www.kwikot.co.za, accessed: Nov 2016.

[116] u blox, "u-blox Cellular Modules, AT commands manual," u-blox, Tech. Rep., Sept 2015. [Online]. Available: https://www.u-blox.com/sites/default/files/u-blox-ATCommands_Manual_(UBX-13002752).pdf